

Article

AutoTuning Environment for Static Obstacle Avoidance Methods Applied to USVs

Rafael Guardeno ^{1,*}, Manuel J. López ¹, Jesús Sánchez ² and Agustín Consegliere ¹

¹ Escuela Superior de Ingeniería, Universidad de Cádiz, 11519 Puerto Real, Spain; manueljesus.lopez@uca.es (M.J.L.); agustin.consegliere@uca.es (A.C.)

² Navantia Sistemas, 11100 San Fernando, Spain; jsanchezpa@navantia.es

* Correspondence: rafael.guardeno@uca.es; Tel.: +34-638-88-70-86

Received: 2 April 2020; Accepted: 22 April 2020; Published: 26 April 2020



Abstract: This work is focused on reactive Static Obstacle Avoidance (SOA) methods used to increase the autonomy of Unmanned Surface Vehicles (USVs). Currently, there are multiple approaches to avoid obstacles, which can be applied to different types of USV. In order to assist in the choice of the SOA method for a particular vessel and to accelerate the pretuning process necessary for its implementation, this paper proposes a new AutoTuning Environment for Static Obstacle Avoidance (ATESOA) methods applied to USVs. In this environment, a new simplified modelling of a LIDAR (Laser Imaging Detection and Ranging) sensor is proposed based on numerical simulations. This sensor model provides a realistic environment for the tuning of SOA methods that, due to its low load computation, is used by evolutionary algorithms for the autotuning. In order to analyze the proposed ATESOA, three SOA methods were adapted and implemented to consider the measurements given by the LIDAR model. Furthermore, a mathematical model is proposed and evaluated for using as USV in the simulation environment. The results obtained in numerical simulations show how the new ATESOA is able to adjust the SOA methods in scenarios with different obstacle distributions.

Keywords: unmanned surface vehicles; autonomous navigation; autotuning environment; obstacle avoidance; LIDAR sensor modelling; vessel modelling; course control; velocity control

1. Introduction

Nowadays, the development of the USV is an active and growing field of research. The reason is that the applications of a USV are very broad, covering from environmental control, as well as many scientific and commercial applications, to national security and surveillance issues [1]. The competencies required a vessel in order to be considered autonomous can be grouped into three fields: Navigation, Guidance and Control, [1–3]. In more detail, these three fields can be decomposed into the subsystems shown in Figure 1. Where, firstly, the state vector of the vessel dynamics must be estimated from state observers and wave filters [4,5]. Once the state of the vehicle is known, the detection system processes the information received by the surrounding sensors in order to generate a model of the environment in which the USV is located [6–10]. Using this model of the environment, the guidance system (composed by algorithms of: obstacle avoidance [8,11–16], path following [4,5,17,18] and path planning [3,7,17,19,20]) demands the course and speed setpoints which guide the vessel safely to its goal. With the aim of ensuring that the vehicle reaches these setpoints, the control system commands the vessel's actuators (propulsion system and steering machine) [4,5]. In this way, the degree of autonomy of the USV is determined by the level of development of each subsystem, as well as by the integration of the set [1–3]. From the subsystems shown in Figure 1, this work is focused on the obstacle avoidance systems applied to USVs [8,10,12–16,21–31]. Most of this work is focussed on the avoidance of dynamic obstacles in open sea situations [8,12–15,21–28]. And, although some of them also consider a specific

approach to static obstacles [15,22,23,25], their study and evaluation is secondary. However, on a lesser scale, works such as [10,16,17,29–31] are centered on the capacity of a USV to avoid static obstacles.

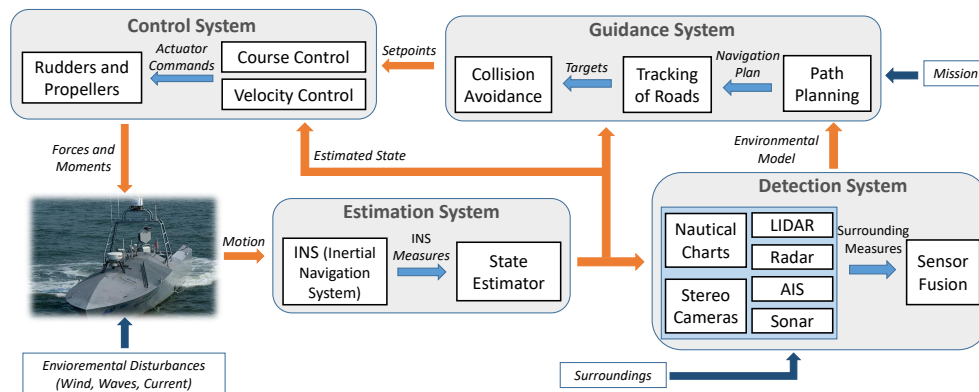


Figure 1. Interaction between the different systems that allow a boat to be considered as a USV.

All these obstacle avoidance methods have a common requirement, they need a simulation environment for their design, development and evaluation. In this aspect, there are very complete approaches [32–35], where the authors are focused on the development of simulation environments for the validation of USVs under realistic sea conditions (wind, current and waves), considering also the models of inertial and LIDARs sensors available in the GAZEBO framework [36]. On the other hand, most of the works focused on obstacle avoidance methods do not take into account the ambient disturbances that affect the USV and, moreover, they assume known: position, dimension and speed of the obstacles. As exceptions, in [12,15] the error present in the estimates of the environment model is taken into account. In addition, in [15,22] the external disturbances that affect the USV are also considered by modelling them as constant forces and moments referred to the earth axes. As an alternative to these extremely simplified simulation environments [10,13,21,23–31], or the highly complete environments proposed in [32–36], in this work a new simulation environment based on the simplified modelling of a LIDAR sensor is proposed. As a result, the time of numerical simulations is reduced and the methods of obstacle avoidance are exposed to a higher level of uncertainty present in the environment surrounding the vessel. This facilitates that the simulation environment proposed in this work can be used by iterative optimization algorithms, such as the evolutionary algorithms [37], with the advantage that it requires less computer load. In this way, the main contribution of this work is achieved, a new autotuning environment for SOA methods applied to USVs. Currently, there are different approaches for the autotuning of unmanned vehicles [38–44]. In [38] the authors use machine learning for autotuning the guidance system on a physical ground vehicle. As autotuning methods applied in the marine environment, in [39] a genetic algorithm is used to autotune the parameters of a sliding mode controller, which is applied to the mathematical model of an oil tanker vessel. On the other hand, the work carried out in [40] is focused on the autotuning of a dynamic positioning system in response to changes in the sea state. This is evaluated in numerical simulations with the dynamic model of an offshore supply vessel. With regard to underwater vehicles, in [41] the authors propose a autotuning method for heading control on an underwater vehicle and validate their approach with the micro-ROV vehicle. In relation to these works, the objective of the ATESOA approach proposed in this paper is to facilitate and accelerate the pretuning process necessary to implement SOA methods in USVs. In addition, in order to evaluate this autotuning environment, three methods of static obstacle avoidance have been adapted and implemented in this work [10,23,29,30], which have already been used in USVs. As a secondary contribution of this work, through the application of the generalized potential fields [45,46], the performance of potential fields method for USVs proposed in [23,30] has been improved. Finally, a USV model must be used to evaluate the new autotuning environment. In the current state of the art different mathematical models of USVs have been proposed [22,26,28,47–49],

most of them for vessels of small length. This is mainly because one of the advantages of this kind of vehicles is that, as these do not need a crew, the size of the vehicle can be reduced, increasing its manoeuvrability and reducing costs [1,2]. Many of these models [22,26,28,47,48] simplify the dynamics of the vessel since they do not take into account the modelling of the actuators and the effect of the ocean current. In order to make realistic the simulation environment proposed in this work, a mathematical model is proposed for an USV of 9.2 m of length. This model has been synthesized from previous work in the field of marine surface vehicle control [4,5,28,49–52], whose parameters have been calculated through numerical simulations in order to obtain a characteristic behavior of marine surface vehicles [5,53–57]. Concretely, the USV model is composed of: model of the vessel and the actuators, current effect and course/speed controllers.

Major Contribution

This work is focused on the reactive obstacle avoidance methods, which form part of USV guidance systems. The main contribution of this work is a autotuning environment for SOA methods applied to USVs, the ATESOA environment. ATESOA can be applied to different SOA methods and vessels. In particular, it is based on the simplified modelling of a LIDAR sensor. This model takes into account the uncertainty present in the environment by considering the perspective effect of the sensor. In addition, a variation of the error present in the measurements delivered is also modelled as a function of the distance to the obstacle. Given its low computational load, this sensor model is used to perform a realistic autotuning of SOA methods using an evolutionary algorithm. In order to analyze the proposed ATESOA, the SOA methods: LROABRA (Local Reactive Obstacle Avoidance Based on Region Analysis), Vector Field Histogram+ and Potential Fields, have been implemented and adapted to the measures delivered by the sensor model. The results obtained have been analysed in detail in terms of performance and robustness.

The organization of this paper is as follows: Section 2 contains the mathematical models that form the new autotuning environment. The Section 3 presents the SOA methods adapted and implemented in this work to evaluate the autotuning environment. In Section 4, the setting of the genetic algorithm used for autotuning is presented, as well as the indicators that have been defined to evaluate the performance of each avoidance method. Section 5 contains the results obtained in numerical simulations. Lastly, in Section 6 the Conclusions are presented.

2. Autotuning Environment Based on Numerical Simulations

This section describes the mathematical models that form the simulation environment proposed in this work for the autotuning of SOA methods applied to USVs.

2.1. USV Model

The USV model is based on the development proposed in [4,5] for the modelling of three degrees of freedom applied to vessels. Specifically, the non-linear modelling proposed by Norrbin [5,49] is taken into account for the damping and the effect of the ocean currents is incorporated into the model in terms of relative velocities, considering that the current variation is significantly slower than the ship dynamics, [5]. Thus, the dynamic model of the vessel is defined as:

$$\begin{aligned}
 \dot{u}(m - X_{\dot{u}}) &= mvr + mx_g r^2 - Y_{\dot{v}} v_r r + X_u u_r + X_{|u|u} |u_r| u_r + \tau_x \\
 \dot{v}(m - Y_{\dot{v}}) + mx_g \dot{r} &= -m u r - X_{\dot{u}} u_r r + Y_v v_r + Y_r r + Y_{|v|v} |v_r| v_r + Y_{|r|v} |r| v_r + Y_{|v|r} |v_r| r + \tau_y \\
 \dot{r}(I_z - N_{\dot{r}}) + mx_g \dot{v} &= -m x_g u r - N_{uv} v_r u_r + N_r r + N_v v_r + N_{|v|r} |v_r| r + N_{|r|v} |r| v_r + N_{|r|r} |r| r + \tau_n
 \end{aligned}$$

$$\underbrace{\begin{pmatrix} \dot{x}_E \\ \dot{y}_E \\ \dot{\psi} \end{pmatrix}}_{\dot{\eta}} = \underbrace{\begin{pmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{pmatrix}}_{R(\psi)} \underbrace{\begin{pmatrix} u \\ v \\ r \end{pmatrix}}_v \quad (1)$$

where η represents the position vector of the vessel referenced to earth axes, v and v_r are the vectors of velocity and relative velocity referenced to body axes, $R(\psi)$ is the rotational matrix from Body-axes reference system to Earth-axes reference system (see Figure 2), $\tau_{act} = [\tau_x \ \tau_y \ \tau_z]^T$ represents the forces and moment applied by the actuators and the viscous non-linear term N_{uv} replaces $(X_{\dot{u}} - Y_{\dot{v}})$, which causes the destabilizing moment or Munk moment [5]. The relative velocity vector of the Model (1) is defined according to Equation (2), where V_c is the module of the current velocity and β_c its direction.

$$v_r = [u_r \ v_r \ r]^T = [(u - V_c \cos(\beta_c - \psi)) \ (v - V_c \sin(\beta_c - \psi)) \ r]^T \quad (2)$$

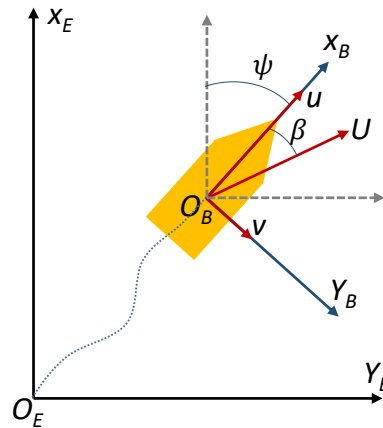


Figure 2. Earth and body axis reference systems adopted in this work.

On the other hand, the actuators are the elements that allow the steering of the vessel [4,5,49–51]. For the USV model proposed in this work, an outboard motor is considered as the actuator, which generates the propulsion force and the steering torque. This actuator is modelled as the set formed by propeller and rudder [49–51]. In this way, the force vector produced by the outboard located on the vessel's centreline is defined according to Equation (3).

$$\tau_{act} = \begin{pmatrix} \tau_x \\ \tau_y \\ \tau_n \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -l_x \end{pmatrix} \begin{pmatrix} T_{act} - D_{act} \\ L_{act} \end{pmatrix}, \quad \begin{aligned} T_{act} &= T_{|n|n}|n|n - T_{|n|u}|n|u_r \\ D_{act} &= D_{|\delta|}|\delta||u_r|u_r \\ L_{act} &= (L_{\delta}\delta - L_{|\delta|\delta}|\delta|\delta)|u_r|u_r \end{aligned} \quad (3)$$

where l_x is the distance of the actuator to the vessel's center of gravity, n represents the revolutions per minute (RPM) of the propeller, δ is the rudder angle relative to the vessel's centerline, T_{act} represents the thrust generated by the propeller and, finally, D_{act} and L_{act} are the drag and transversal forces generated by the rudder, respectively. In addition, the dynamics of the outboard motor is taken into account as:

$$\begin{aligned} \dot{\delta} &= (1/\tau_{\delta})(\delta_c - \delta), & \delta &\in [-\delta_{lim}, \delta_{lim}], & \dot{\delta} &\in [-\dot{\delta}_{lim}, \dot{\delta}_{lim}] \\ \dot{n} &= (1/\tau_n)(n_c - n), & n &\in [n_{min}, n_{max}] \end{aligned} \quad (4)$$

where δ_c and n_c represent the setpoints for the rudder angle and the propeller RPM, respectively. The time constants τ_{δ} and τ_n define the dynamics of the outboard motor.

Once the mathematical model has been defined, in order to adjust its parameters, the turning radius ($K_r = \sqrt{Iz/m}$) and the Nomoto model identified in [28] for the vessel *Viknes 830*, as well as its maximum speed, have been taken as references to synthesize the proposed mathematical model and its parameter values. The main features of *Viknes 830* are listed in Table 1.

The parameters of the USV model proposed in this paper are collected in Table 2. Assuming a homogeneous mass distribution and taking as a reference the *Viknes 830*, the mass of the new model

and its moment of inertia are obtained according to the Equation (5). In addition, the added masses that affect the vessel are estimated according to [52], see Equation (6).

$$m = \left(\frac{LBD}{L_v B_v D_v} \right) m_v, \quad I_z = K_{r_v}^2 m \quad (5)$$

where L , B y D represent the length, the beam and the draft of the vessel, respectively.

$$X_{\dot{u}} = -0.05m, \quad Y_{\dot{v}} = -(1/2)\rho\pi D^2 L, \quad N_{\dot{r}} = -(1/24)(0.1mB^2 + \rho\pi D^2 L^3) \quad (6)$$

where ρ defines the density of seawater.

Table 1. *Viknes 830* parameters expressed in the international system of units [28].

Symbol	Description	Value	Symbol	Description	Value
B_v	Beam	2.97	m_v	Mass	3980
L_v	Length	8.52	I_{z_v}	z-axis moment of inertia	19,703
D_v	Draft	0.82	u_{max}	Maximum forward speed	10.5
$K_{Nomoto-Model}$	Gain r/δ	0.5	$\tau_{Nomoto-Model}$	Time constant	4
r_{max}	Maximum angular rate	0.35	δ_{max}	Maximum rudder angle	0.26

The other parameters are adjusted to obtain stationary values for forward speed and angular rate close to those shown in Table 1. In order to verify this setting, a zero current velocity is assumed ($V_c = 0$) and, once the operating range of the actuators (n_c and δ_c) has been discretised, the points of equilibrium of the system are obtained (stationary values of the velocity vector v). As can be seen in Figure 3, the forward speed increases with propoller revolution (where $u_{max} = 10.5$ m/s) and falls when the rudder is engaged. Furthermore, the relation of the stationary value of r with the rudder angle is consistent with the behaviour of surface marine vehicles exposed in [5,53–56].

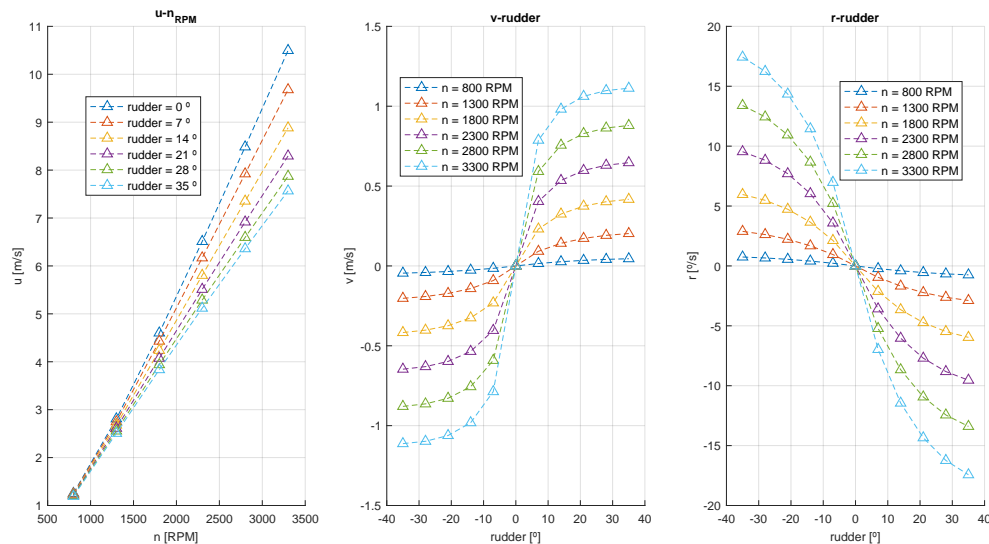


Figure 3. Equilibrium points or stationary velocities of the Model (1), which have been obtained by discretizing its operating range.

On the other hand, the parameters listed in Table 2 must ensure a stable behavior of the Model (1) at all its operating points. In order to check its dynamic behavior, Model (1) is linearized at each operating point shown in Figure 3. As a result, the effective time constants obtained for the velocity vector v are shown in Figure 4. As can be seen, the system is stable over its entire operating range and its dynamics becomes faster in function of the vessel's speed [5,53–56]. Moreover, with a zero rudder

angle, the effective time constant τ_r varies, depending of the speed, around the 4 s obtained by [28] in the Nomoto model identified for the *Viknes 830*. To conclude with the analysis of the Model (1), it is evaluated through numerical simulations by carrying out different sea tests [57]. The Runge-Kutta numerical integration method of fourth order is used with an integration step of 0.01 s, holding $n_c = 2430$ RPM and $V_c = 0$ m/s. The Turning Circle Tests can be seen in Figure 5a. On the other hand, in Figure 5b,c the spiral (or Dioudonne) manoeuvre and zig-zag manoeuvres are shown.

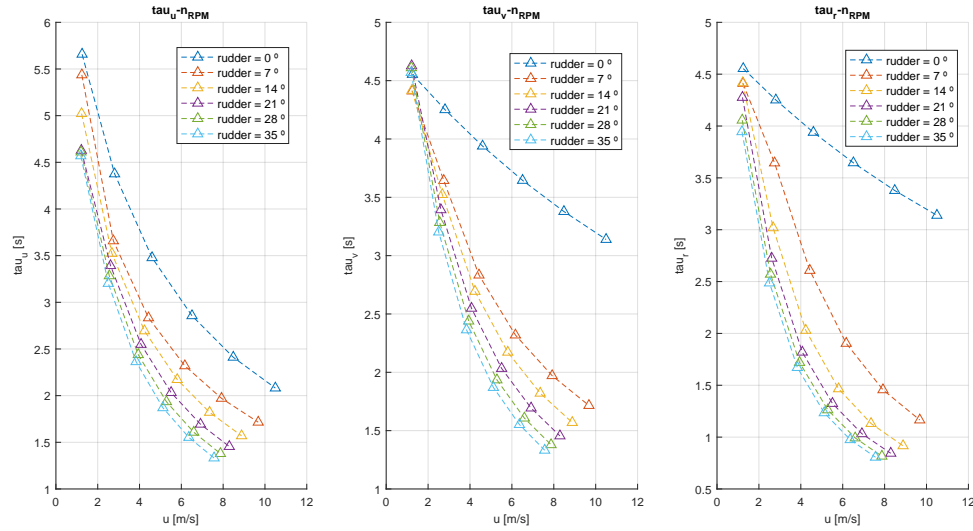


Figure 4. Variation of the effective time constants of the Model (1) linearized at different points of its operating range.

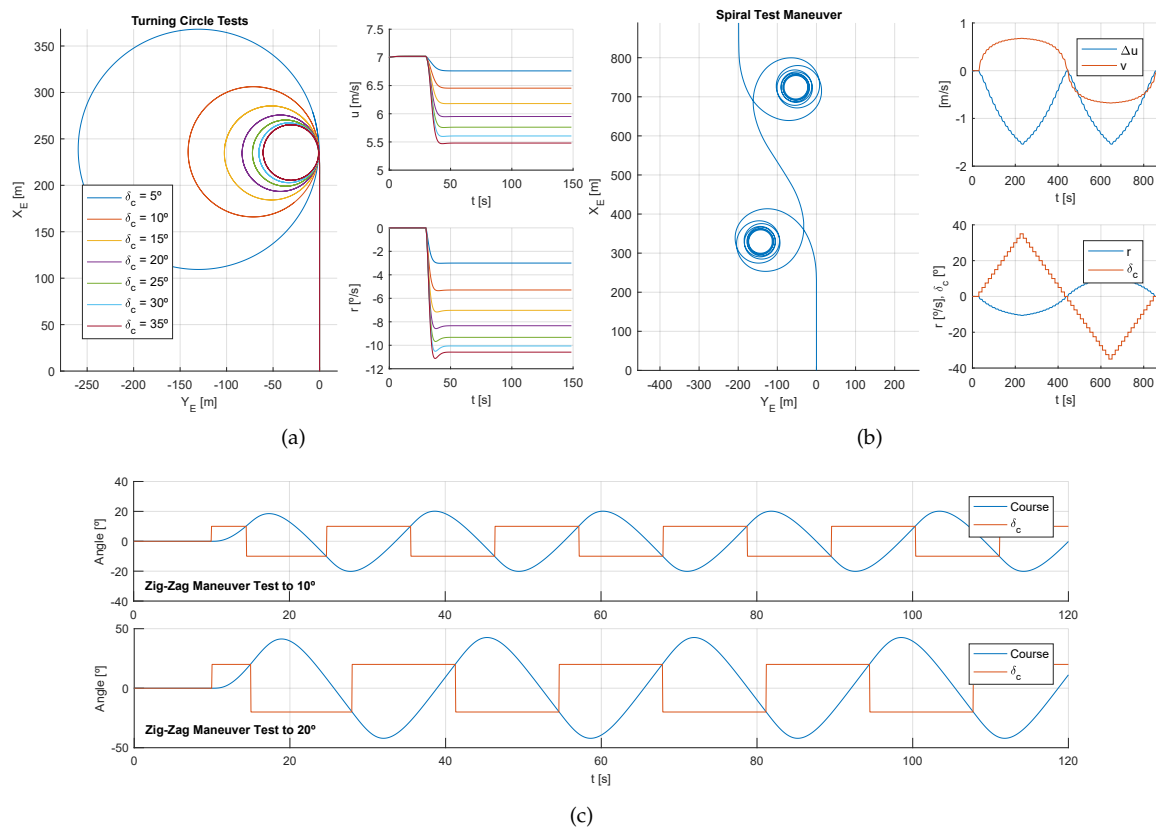


Figure 5. Numerical simulations of the Model (1) which is evaluated on different sea tests: (a) Turning Circle Tests. (b) Spiral maneuvering. (c) Zig-zag maneuvers. The propeller is kept at $n_c = 2430$ RPM.

Table 2. Parameters of the vessel model proposed in this work.

Symbol	Value	Symbol	Value	Symbol	Value	Symbol	Value
L	9.2 m	B	3 m	D	0.7 m	x_g	0.5 m
l_x	4.5 m	m	3705.78 kg	I_z	18,345.50 kg m ²	$X_{\dot{u}}$	−185.23 kg
$Y_{\dot{v}}$	−7272.34 kg	N_r	−51,433.21 kg m ²	ρ	1027 kg/m ³	X_u	−512.5
Y_v	−2200	Y_r	−500	N_r	−15,000	N_v	−311.1
$X_{ u u}$	−59.5	$Y_{ v v}$	−5000	$Y_{ r v}$	−7000	$N_{ r r}$	−4000
$N_{ v r}$	−50	$Y_{ v r}$	−40	$N_{ r v}$	−1700	N_{uv}	−22
$T_{ n n}$	1.2×10^{-3}	$T_{ n u}$	3.27×10^{-2}	L_δ	1.3886	$L_{ \delta \delta}$	1.28×10^{-2}
$D_{ \delta }$	0.71	τ_n	0.5 s	τ_δ	1 s	n_{max}	3300 RPM
n_{min}	800 RPM	δ_{lim}	35°	$\dot{\delta}_{lim}$	10°/s		

2.2. Course and Speed Controllers

Most obstacle avoidance systems for USVs [10,12,13,15,21–27,29–31] require that the vessel be equipped with course and speed controllers. These controllers guarantee, within a range of defined sea states, that the vessel reaches the course and speed requested by the avoidance system in a finite time. To this end, there are currently a multitude of control methods [4,5,52,53,58]. As the objective of this work is not focused on the design of the controllers, and in order to limit the problem, two PID structures are used to control the course and speed, see Equation (7). Moreover, due to the strong non-linearity of the Model (1), the adaptive control technique Gain Scheduling is used [53,58]. For this purpose, the controller parameters are set experimentally for three operating points: low ($u_{low} = 3$ m/s), medium ($u_{medium} = 6$ m/s) and high ($u_{high} = 9$ m/s) speed, see Table 3. For intermediate speeds, linear interpolations are carried out between the PIDs. Finally, these controllers have been discretised using the Euler backwards method with a sampling period $T_m = 0.05$ s.

$$\begin{aligned}
 \delta_c(k) &= \underbrace{K_{P_\psi} e_\psi(k)}_{\delta_P(k)} + \underbrace{T_m K_{I_\psi} e_\psi(k)}_{\delta_I(k)} + \underbrace{\delta_I(k-1) + \frac{K_{D_\psi}}{T_m} (e_{f_\psi}(k) - e_{f_\psi}(k-1))}_{\delta_D(k)} \\
 n_c(k) &= \underbrace{K_{P_u} e_u(k)}_{n_P(k)} + \underbrace{T_m K_{I_u} e_u(k)}_{n_I(k)} + \underbrace{n_I(k-1) + \frac{K_{D_u}}{T_m} (e_{f_u}(k) - e_{f_u}(k-1))}_{n_D(k)} \\
 e_{f_i}(k) &= \frac{1}{c_f + 1} (e_i(k) + c_f e_{f_i}(k-1)), \quad e_i(k) = sp_i(k) - i(k)
 \end{aligned} \tag{7}$$

where $i \in \{u, \psi\}$, $c_f = \tau_f / T_m$ and the time constants have been fixed as $\tau_f = 0.1(K_D / K_P)$.

Table 3. Parameters of the course and speed controllers.

	K_{P_ψ}	K_{I_ψ}	K_{D_ψ}	K_{P_u}	K_{I_u}	K_{D_u}
u_{low}	−40	−0.01	−30	40	2	6
u_{medium}	−16	−0.01	−32	35	2	12
u_{high}	−9.5	−0.01	−17	45	2	18

Due to the effect of external disturbances, a compensation is carried out on the error signals of the controllers (7) [5,52]. In this way, the course angle ($\chi = \psi + \beta$, see Figure 2) is controlled instead of the heading angle (ψ), as well as the module of the velocity vector (U) of the USV.

$$\begin{aligned}
 e_\psi(k) &= sp_\psi(k) - \chi(k), \quad \chi(k) = \psi(k) + \text{atan2}(v(k), u(k)) \\
 e_u(k) &= sp_u(k) - U(k), \quad U(k) = \sqrt{u(k)^2 + v(k)^2}
 \end{aligned} \tag{8}$$

2.3. Simplified Modelling of a LIDAR Sensor

As a novel contribution made in this work, and with the aim of increasing the realism of the numerical simulations carried out, a LIDAR sensor has been modelled. As a reference, the characteristics of the LIDAR Ultra Puck of the company Velodyne have been used [59]. In addition, as in [10,12,13,15,21–31], a 2D Cartesian space (x_E, y_E) is used to evaluate the obstacle avoidance systems applied to USVs. Therefore, only one of the vertical channels of the LIDAR sensor is taken into account. Thus, the computation of the sensor model is decreased, reducing simulation times and allowing its use in iterative optimization algorithms, as evolutionary algorithms. In particular, from the information provided by the manufacturer in [59], the following data are taken:

- Distance Measurement Range of 200 m ($d_{range} = 200$ m).
- Horizontal Field of View of 360° ($h_{range} = 360^\circ$).
- Horizontal Angular Resolution between 0.1° and 0.4° . Looking for the worst case, $h_{res} = 0.4^\circ$.
- Rotation Rate between 5 Hz to 20 Hz. Looking for a balance between computation and refresh rate of the environment information (T_m^L), a rotation frequency $f_m^L = 10$ Hz is set.
- Range Accuracy up to ± 3 cm. The minimum value for the standard deviation of the WGN (White Gaussian Noise) that affects the measurement is taken as $\sigma_{min}^L = 0.03$ m.

Before proceeding with the modelling of the LIDAR sensor, it is necessary to define the obstacle scenarios in which the USV is located. In this work, each obstacle $Obst_E$ is defined as a polygon formed by four concatenated segments, which are referenced to Earth-axes. Consequently, a scenario (S_E) is the result of the union of a finite number of obstacles (n_O), see Equation (9).

$$S_E = \bigcup_{i_O=1}^{n_O} Obst_E^{i_O}, \quad Obst_E^{i_O} = \left\{ \begin{array}{l} seg_{1E}^{i_O} = (x_{1E}^{i_O}, y_{1E}^{i_O}, x_{2E}^{i_O}, y_{2E}^{i_O}), \quad seg_{2E}^{i_O} = (x_{2E}^{i_O}, y_{2E}^{i_O}, x_{3E}^{i_O}, y_{3E}^{i_O}), \\ seg_{3E}^{i_O} = (x_{3E}^{i_O}, y_{3E}^{i_O}, x_{4E}^{i_O}, y_{4E}^{i_O}), \quad seg_{4E}^{i_O} = (x_{4E}^{i_O}, y_{4E}^{i_O}, x_{1E}^{i_O}, y_{1E}^{i_O}) \end{array} \right\} \quad (9)$$

where $seg_{1E...4E}^{i_O}$ represents the concatenated segments that form each obstacle and $x_{1E...4E}^{i_O}$ are the coordinates of its four points.

It is important to highlight that we have chosen to model each obstacle as a rectangle due to the facility for generating scenarios from these polygons. However, this LIDAR sensor model can also be used in the presence of irregular polygons such as the ones that define the marine environments [16,17,19]. For this purpose, two approaches can be taken. Firstly, Delaunay Triangulation [60] can be applied to polygons that form the coastlines [61]. This triangulation decomposes the polygon into a finite number of triangles (n_O). Thus, each triangle is defined as a polygon formed by three concatenated segments. Secondly, the scenario can be defined as the set of segments collected in .shp files such as the ones collected in [61].

Defined the LIDAR parameters, as well as the obstacle scenario (S_E), the measurement of this sensor can be modelled as the set of segments L_B (10), which is referenced to Body-axes.

$$L_B = \left\{ seg_{8B}^{i_L} (x_{1B}^{i_L}, y_{1B}^{i_L}, x_{2B}^{i_L}, y_{2B}^{i_L}) \mid x_{1B}^{i_L} = y_{1B}^{i_L} = 0, \quad x_{2B}^{i_L} = \cos(i_L h_{res}) d_{range}, \quad y_{2B}^{i_L} = \sin(i_L h_{res}) d_{range} \right\} \quad (10)$$

$$I_L = \{i_L \mid i_L \in \mathbb{N}, i_L < n_L\}, \quad n_L \in \mathbb{N}$$

where $x_{1B,2B}^{i_L}$ and $y_{1B,2B}^{i_L}$ represent the coordinates of the two points that form a segment and n_L is the number of segments (LIDAR beams) that compose the set L_B ($n_L = h_{range}/h_{res} = 900$).

In order to model the distances measured by the LIDAR in its horizontal field of view (h_{range}), the points of intersection between the two sets of segments S_E and L_B are taken. For this purpose, both sets of segments must refer to the same reference system. Therefore, once the set L_B has been expressed in matrix form, it is rotated and translated to Earth-axes at each simulation instant (k):

$$\underbrace{\begin{pmatrix} x_{1E}^0 & y_{1E}^0 & x_{2E}^0 & y_{2E}^0 \\ \vdots & \vdots & \vdots & \vdots \\ x_{1E}^{i_L} & y_{1E}^{i_L} & x_{2E}^{i_L} & y_{2E}^{i_L} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1E}^{n_L-1} & y_{1E}^{n_L-1} & x_{2E}^{n_L-1} & y_{2E}^{n_L-1} \end{pmatrix}}_{L_E} = \underbrace{\begin{pmatrix} x_{1B}^0 & y_{1B}^0 & x_{2B}^0 & y_{2B}^0 \\ \vdots & \vdots & \vdots & \vdots \\ x_{1B}^{i_L} & y_{1B}^{i_L} & x_{2B}^{i_L} & y_{2B}^{i_L} \\ \vdots & \vdots & \vdots & \vdots \\ x_{1B}^{n_L-1} & y_{1B}^{n_L-1} & x_{2B}^{n_L-1} & y_{2B}^{n_L-1} \end{pmatrix}}_{L_B} \begin{pmatrix} R_p^T & \mathbf{0} \\ \mathbf{0} & R_p^T \end{pmatrix} + \begin{pmatrix} 1 & \dots & 1 & \dots & 1 \end{pmatrix}^T \begin{pmatrix} x_E(k) & y_E(k) & x_E(k) & y_E(k) \end{pmatrix}, \quad R_p = \begin{pmatrix} \cos(\psi(k)) & -\sin(\psi(k)) \\ \sin(\psi(k)) & \cos(\psi(k)) \end{pmatrix} \quad (11)$$

The set of segments that form the scenario is also expressed in matrix form S_E :

$$S_E = \begin{pmatrix} Obst_E^1 \\ \vdots \\ Obst_E^{i_O} \\ \vdots \\ Obst_E^{n_O} \end{pmatrix}, \quad Obst_E^{i_O} = \begin{pmatrix} seg_{1E}^{i_O} \\ seg_{2E}^{i_O} \\ seg_{3E}^{i_O} \\ seg_{4E}^{i_O} \end{pmatrix} = \begin{pmatrix} x_{1E}^{i_O} & y_{1E}^{i_O} & x_{2E}^{i_O} & y_{2E}^{i_O} \\ x_{2E}^{i_O} & y_{2E}^{i_O} & x_{3E}^{i_O} & y_{3E}^{i_O} \\ x_{3E}^{i_O} & y_{3E}^{i_O} & x_{4E}^{i_O} & y_{4E}^{i_O} \\ x_{4E}^{i_O} & y_{4E}^{i_O} & x_{1E}^{i_O} & y_{1E}^{i_O} \end{pmatrix} \quad (12)$$

As the problem of the intersection of segments can be solved from the field of computational geometry [60], in this work the matrices (11) and (12) are used as inputs for the function proposed in [62]. This function calculates the intersections between two sets of segments, considering coincident and parallel segments. Between other outputs, it returns the matrices F_{Inter} and ND_{L_E} , both of dimension $n_L \times n_O$. Where F_{Inter} defines which segments of the sets (11) and (12) are intersected ($F_{Inter}(i_L, i_O) = 1$) and ND_{L_E} returns the normalized distances between the intersection points and the starting points of the segments of the set L_E . From the matrix F_{Inter} , in the Equation (13) the vector f_{Inter}^L is obtained, which indicates that beams of the LIDAR sensor have detected an obstacle.

$$f_{Inter}^L = \left(f_{Inter}^0 \quad \dots \quad f_{Inter}^{i_L} \quad \dots \quad f_{Inter}^{n_L-1} \right)^T, \quad f_{Inter}^{i_L} = \begin{cases} 1 & \text{if } F_{Inter}^{i_L} \cdot F_{Inter}^{i_L^T} \neq 0 \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

$$F_{Inter}^{i_L} \in M_{1 \times n_O}(\mathbb{R})$$

Once the beams that have detected obstacles are known, the measured distance vector D_m^L is defined by the Equation (14). This LIDAR modelling considers the perspective effects that affect to this kind of sensors [1,21,26,32], since from each beam only the closest distance ($d_{min}^{i_L}$) is taken, while the later points of intersection with other segments (obstacles) are ignored.

$$D_m^L = \left(d_m^0 \quad \dots \quad d_m^{i_L} \quad \dots \quad d_m^{n_L-1} \right)^T, \quad d_m^{i_L} = \begin{cases} d_{min}^{i_L} + \eta^{i_L} & \text{if } f_{Inter}^{i_L} = 1 \\ d_{range} & \text{otherwise} \end{cases} \quad (14)$$

$$d_{min}^{i_L} = \min \left(ND_{L_E}^{i_L} \right), \quad ND_{L_E}^{i_L} \in M_{1 \times n_O}(\mathbb{R})$$

where η^{i_L} is the WGN that affects to each distance measurement, whose standard deviation (σ^{i_L}) depends of $d_{min}^{i_L}$ according to the Equation (15). Thus, the accuracy of each measure of the sensor is modelled with continuous limits and depends on the distance to the obstacle.

$$\sigma^{i_L} = \left(\tanh \left(\frac{10d_{min}^{i_L}}{d_{range}} - 5 \right) + 1 \right) \left(\frac{\sigma_{max}^L - \sigma_{min}^L}{2} \right) + \sigma_{min}^L \quad (15)$$

where the parameter σ_{max}^L must be set according to the sea states. Thereby, the performance of the sensor depends on the distance of the obstacles as well as of the environmental conditions. As maximum noise level, in this work $\sigma_{max}^L = 5$ m has been set.

Together with the measured distance matrix (D_m^L) and intersection flags (f_{Inter}^L), the sensor model also provides the matrix of detected points (16). Figure 6 shows the measurements delivered by the modelling proposed in this work for a LIDAR sensor.

$$P_{obstB}^L = \begin{pmatrix} x_{obstB}^0 & \cdots & x_{obstB}^{i_L} & \cdots & x_{obstB}^{n_L-1} \\ y_{obstB}^0 & \cdots & y_{obstB}^{i_L} & \cdots & y_{obstB}^{n_L-1} \end{pmatrix}^T, \quad \begin{aligned} x_{obstB}^{i_L} &= \cos(i_L h_{res}) d_m^{i_L} \\ y_{obstB}^{i_L} &= \sin(i_L h_{res}) d_m^{i_L} \end{aligned} \quad (16)$$

where the reference system of the LIDAR sensor coincides with the Body-axes reference system.

This modelling, with respect to other works [10,12,13,15,21–31] that also evaluate avoidance systems for USVs through numerical simulations, is a novel contribution. Because it exposes the avoidance systems to a higher level of the uncertainty present in the environment surrounding the vessel, without reaching the complexity required when a complete modelling of the environment is realized [32–36]. This facilitates its application in autotuning environments based on evolutionary algorithms, due to the large number of simulations that these algorithms must carry out. As a visual demonstration, the video [63] shows the LROABRA algorithm adapted to the approach for the LIDAR sensor model proposed in this work. As can be seen in this video, this modelling adds measurement error to the detected points in function of their distance to the vehicle. In addition, this model also takes into account the uncertainty present in the environment, since it does not detect hidden obstacles due to the effects of perspective.

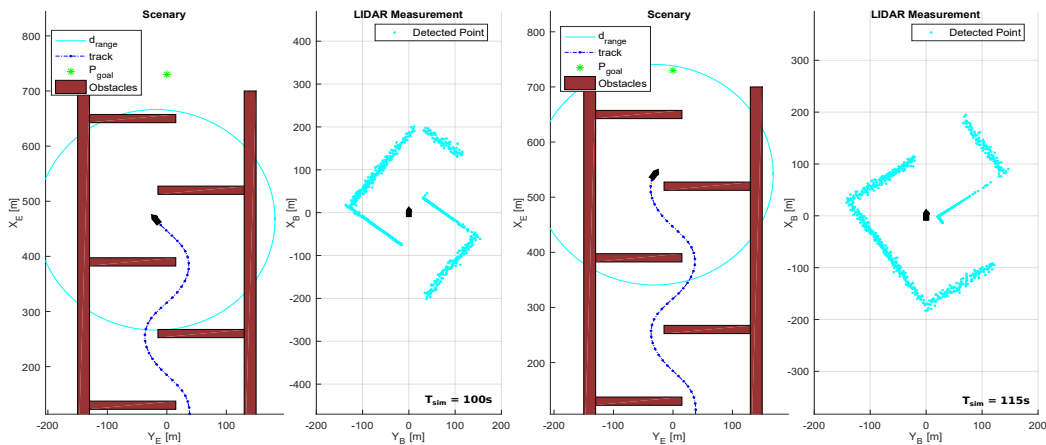


Figure 6. Point detected by the LIDAR Model (16) during different simulation moments T_{sim} .

2.4. Scenarios for Static Obstacle Avoidance Methods Testing

Before the description of the obstacle scenarios, it is necessary to emphasize that the SOA algorithms studied in this work are reactive methods [11]. These methods are characterized by the use of environmental information processed in real time from the measurements delivered by different sensors. Therefore, these methods do not ensure that the vehicle reaches the target if there are local minima. For this reason, in order to provide a complete solution to the problem of autonomous navigation [1–3], it is necessary to combine obstacle avoidance methods or reactive methods [8,10,12–15,21–31] with global or path planning algorithms [3,7,17,19,20]. In this sense, the obstacle scenarios studied in this work are focused only on the reactive part, unknown scenarios that the autonomous vehicle discovers while navigating. Without considering scenarios with local minimums such as those that can be found in marine environments (in which it would be necessary to have a guidance system that also incorporates a global algorithm). Specifically, the scenarios used to carry out the autotuning and to evaluate the SOA methods studied in this work are shown in Figure 7. A scenario is defined by the obstacles (9), the goal

point P_{goal} and the initial position and speed vectors (η, v_r) of the USV. In this work, five scenarios are proposed. With scenario 1 the implementation of the SOA methods is verified. The second scenario, mainly, is used to evaluate the stability of the avoidance algorithm in situations where there are close obstacles without risk of collision. As well as the subsequent response of these methods when a collision risk is presented. Scenario 3 places the USV in a zig-zag scenario where all directions, at greater or lesser distances, lead to collision. Thus, the behaviour of SOA methods in closed environments is evaluated. Finally, scenarios 4 and 5 are densely occupied and allow a wide range of alternative routes. These scenarios are used to study the robustness of SOA methods that, while these have been adjusted for a different scenario, must solve new geometric situations not contemplated in their initial adjustment. In all scenarios the vehicle starts moving at the goal speed $u_{goal} = u = 7$ m/s and, looking for the most unfavourable case in which the current leads the USV towards the obstacles, $V_c = 0.5$ m/s and $\beta_c = 0^\circ$ has been set.

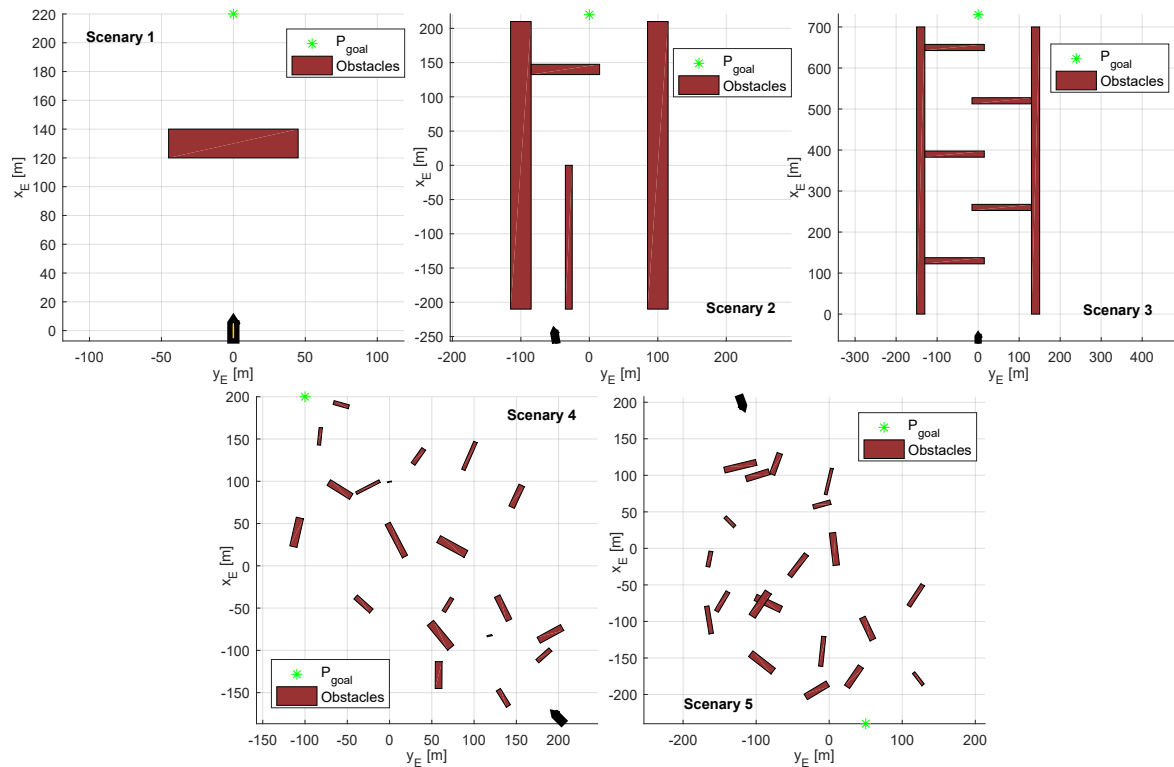


Figure 7. Obstacle scenarios used in this work for autotuning and evaluation of SOA methods.

3. Static Obstacle Avoidance Methods

In order to evaluate the new autotuning environment proposed in this work, three obstacle avoidance methods have been adapted and implemented to the LIDAR sensor Model (10).

3.1. LROABRA Method

The LROABRA method is proposed in [29] as a static obstacle avoidance system that allows USVs to navigate at high speeds. In this approach, the obstacles are represented in a system of polar coordinates referenced to Body-axes, where each obstacle is modelled as a circle of radius r_i and position $O_i(\rho_i, \theta_i)$. From these circles, the non-permitted directions are defined:

$$B_{obs} = \bigcup_{i=1}^{n_O} R_{osb}^i, \quad R_{osb}^i = \begin{cases} [\psi(k) + \theta_i - \phi_i - \sigma, \psi(k) + \theta_i + \phi_i + \sigma] & \text{if } \rho_i \leq d_{near} \\ \emptyset & \text{otherwise} \end{cases} \quad (17)$$

where σ is a safety angle used to expand the angular space occupied by obstacles, d_{near} is the threshold distance from which the obstacles are considered and $\phi_i = \arcsin(r_i/\rho_i)$.

Based on [64], the guidance directions (θ) and speed commands (sp_u) that can be reached by the USV are limited to the following dynamic window:

$$\begin{aligned} V_T &= \{sp_u \mid sp_u \in [u(k) - \dot{u}_{max}T_m^a, u(k) + \dot{u}_{max}T_m^a]\} \\ V_{Head} &= \left\{ \theta \mid \theta \in \left[\psi(k) + r(k)T_m^a - \frac{1}{2}\dot{r}_{max}T_m^{a^2}, \psi(k) + r(k)T_m^a + \frac{1}{2}\dot{r}_{max}T_m^{a^2} \right] \right\} \end{aligned} \quad (18)$$

where \dot{u}_{max} and \dot{r}_{max} represent the maximum linear and angular acceleration of the vessel, respectively, and T_m^a is the sample time of the algorithm.

Once the sets of forbidden (B_{obs}) and reachable (V_{Head}) orientations have been defined, the Heuristics (19) is maximized in order to obtain the angle $\theta_{optimal}$.

$$\begin{aligned} F_\epsilon(\theta) &= \underbrace{\epsilon \left(1 - \frac{|\theta_{goal} - \theta|}{\pi} \right)}_{f_{head}(\theta)} + (1 - \epsilon) \underbrace{\left(\frac{1}{n_O} \sum_{i=1}^{n_O} \frac{|\theta - (\theta_i + \psi(k))|}{\pi} \right)}_{f_{safe}(\theta)} \\ \theta &\in V_{Head}, \quad \theta \notin B_{obs}, \quad \theta \in [-\pi, \pi] \end{aligned} \quad (19)$$

where ϵ is the heuristic adjustment parameter and θ_{goal} represents the goal direction.

Due to the fact that the angle $\theta_{optimal}$ defines a safe and reachable direction for the USV, it is used to insert a temporary guidance waypoint $P_{insert}(x_{insert}, y_{insert})$ at a distance D_{insert} .

$$\begin{aligned} x_{insert} &= x_E(k) + D_{insert}\cos(\theta_{optimal}) \\ y_{insert} &= y_E(k) + D_{insert}\sin(\theta_{optimal}) \end{aligned}, \quad sp_\psi(k) = \text{atan2}(y_{insert} - y_E(k), x_{insert} - x_E(k)) \quad (20)$$

where P_{insert} is updated when the USV reaches it or if the guidance angle is far from the optimal angle:

$$\begin{aligned} |x_{insert} - x_E(k), y_{insert} - y_E(k)| &\leq d_{gate} \\ |sp_\psi(k) - \theta_{optimal}(k)| &\geq \theta_{gate} \end{aligned} \quad (21)$$

where d_{gate} and θ_{gate} represent the linear and angular distance thresholds that set the P_{insert} update.

On the other hand, the speed setpoint sp_u is calculated according to the Equation (22), which is limited to the dynamic window V_T (18).

$$sp_u(k) = u_{goal}\eta_L(k) \left(1 - \frac{|\theta_\delta(k)|}{\pi} \right), \quad \theta_\delta(k) = sp_\psi(k) - \psi(k) \quad (22)$$

where the variable η_L depends of the minimum distance to obstacles which lie within the impact zone of the USV, see Equation (23).

$$\eta_L = \begin{cases} \eta_1 = \frac{r(k)}{\max(\theta_\delta(k), r(k))} & \text{if } d_{min} \leq d_{near} \\ \eta_2 = \frac{d_{min}^2 + 2r_{max}d_{min}}{d_{near}^2 + r_{max}d_{near}} & \text{otherwise} \end{cases}, \quad d_{min} = \min \left\{ d_{obs}^i \leq d_{near} \wedge |\theta_i| < \theta_v \right\} \quad (23)$$

where r_{max} is the maximum angular rate of the vessel, d_{obs}^i is the distance to the center of each obstacle and θ_v defines the angular impact zone of the USV.

In order to expose the LROABRA algorithm to a higher level of uncertainty in the environment, in this work the set of non-permitted directions is modified. Thus, B_{obs} is generated from the measurements delivered by the LIDAR sensor model, considering each beam from the Model (10) as an obstacle, see Equation (24). Furthermore, with the purpose that the algorithm considers the direction of the

velocity vector, in the Equations (18), (19) and (22), $\psi(k)$ is replaced by $\chi(k)$. One of the executions of the LROABRA algorithm implemented in this work is shown in Figure 8.

$$B_{obs} = \bigcup_{i=1}^{n_L} R_{osb}^i, \quad R_{osb}^i = \begin{cases} [\chi(k) + i_L h_{res} - \sigma, \chi(k) + i_L h_{res} + \sigma] & \text{if } d_m^{i_L} \leq d_{near} \\ \emptyset & \text{otherwise} \end{cases} \quad (24)$$

where the term $\theta_i \pm \phi_i$ of the Equation (17) has been replaced by $i_L h_{res}$.

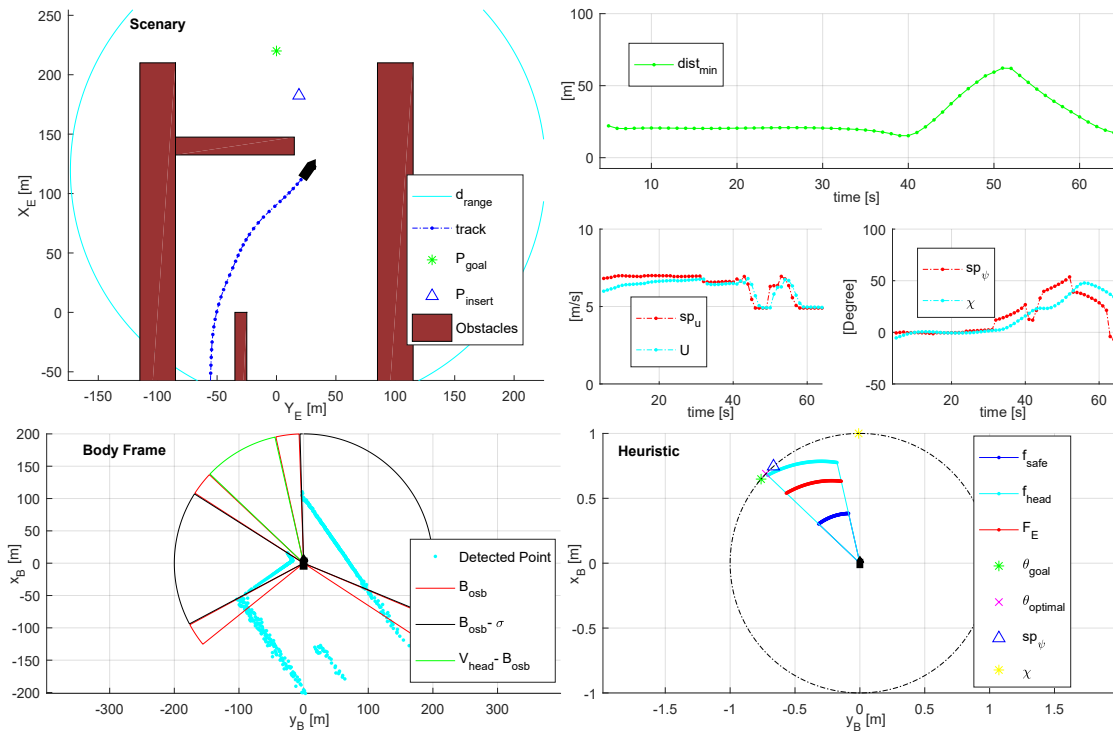


Figure 8. Implementation of the LROABRA algorithm modified for the simulation environment for USVs proposed in this work.

3.2. Potential Fields Methods

The potential field (PF) method, due to its simplicity, has been widely used in different problems of mobile robotics [7,11,16,19,23,26,30,45,65]. A disadvantage of the classical potential fields used as a reactive obstacle avoidance method is that it can produce oscillatory trajectories [46]. In order to improve their performance in USVs, the classical potential fields [65] are modified by different authors in [23,30]. In their work, the forces of attraction and repulsion are modelled according to Equation (25).

$$f_{att} = m_{att} \alpha_{att} |P_{goal} - P_{USV}(k)|^{m_{att}-1}$$

$$f_{rep}^i = \underbrace{\eta_{rep} \left(\frac{1}{p_s^i} - \frac{1}{p_o} \right) \frac{1}{p_s^{i2}} |P_{USV}(k) - P_{goal}|^{n_{rep}}}_{f_{rep1}^i} + \underbrace{\frac{n_{rep}}{2} \eta_{rep} \left(\frac{1}{p_s^i} - \frac{1}{p_o} \right)^2 |P_{USV}(k) - P_{goal}|^{n_{rep}-1}}_{f_{rep2}^i} \quad (25)$$

where $P_{USV}(k) = [x_E(k) \ y_E(k)]$, P_{goal} is the position of the goal waypoint, α_{att} and m_{att} are positive parameters used to tune the magnitude of the attraction potential, p_o is the distance that limits the range of influence of the obstacles, p_s^i is the distance to each obstacle, and, finally, the magnitude of the repulsion potential is tuned with the positive parameters η_{rep} and n_{rep} .

The repulsive forces of Equation (25), as shown in [23], are generated from the vertices of the obstacles which form the coastline. In addition, the distances considered in [23] are in the range

of kilometres. In order to expose the algorithm to a higher level of uncertainty in the environment and evaluate it in proximity situations ($d_{range} = 200$ m), which increases the difficulty of avoidance scenarios, in this work a repulsive force is generated by each of the beams of the LIDAR sensor (10). In this way, the total repulsive force, the attraction force and the resulting force, all of them referenced to Body-axes, are defined according to Equation (26).

$$\begin{aligned} F_{rep}(k) &= \sum_{i_L=1}^{n_L} f_{rep}^{i_L} \begin{pmatrix} \cos(i_L h_{res} - \zeta(k)) & \sin(i_L h_{res} - \zeta(k)) \end{pmatrix} \\ F_{att}(k) &= f_{att} \begin{pmatrix} \cos(\theta_{goal}(k) - \chi(k)) & \sin(\theta_{goal}(k) - \chi(k)) \end{pmatrix} \\ F_R(k) &= F_{rep}(k) + F_{att}(k) = (F_{Rx} \ F_{Ry}) \end{aligned} \quad (26)$$

where $\zeta(k) = \psi(k) - \chi(k)$, $\theta_{goal}(k) = \text{atan2}(y_{goal} - y_E(k), x_{goal} - x_E(k))$ and the module of each repulsive force ($f_{rep}^{i_L}$) corresponds to term f_{rep}^i of Equation (25), in which p_s^i has been replaced by the measured distance $d_m^{i_L}$, which is defined in Equation (14).

The course setpoint sp_ψ is obtained from the resulting force (F_R) according to the Equation (27), [46]. As can be seen, a low-pass filter is applied to reduce the oscillations [66].

$$\begin{aligned} \theta_{FR}(k) &= \text{atan2}(F_{Ry}, F_{Rx}) + \chi(k) \\ sp_\psi(k) &= \frac{1}{c_F + 1} (\theta_{FR}(k) + c_F sp_\psi(k-1)) \end{aligned} \quad (27)$$

where $c_F = \tau_F / T_m^a$, τ_F is the effective time constant of the filter applied to the resulting force and T_m^a is the sample time of the obstacle avoidance method.

On the other hand, in [23,30] the authors do not consider changes in the speed of the vessel. In order to ensure that the potential fields have the same manoeuvrability over the vessel as the other SOA methods studied in this paper, the approach proposed in [66] is adopted to calculate sp_u . Thus, the speed setpoint depends on the resulting force and the speed of the vessel, see Equation (28).

$$\begin{aligned} sp_u(k) &= u_{goal}(1 - |\cos(\theta)|) \\ \cos(\theta) &= \begin{cases} 0 & \text{if } |F_R| = 0 \\ \frac{uF_{Rx} + vF_{Ry}}{|U_B||F_R|} & \text{otherwise} \end{cases} \end{aligned} \quad (28)$$

where $U_B(u, v)$ represents the vessel's velocity vector referenced to Body-axes.

As will be shown in the results section, despite the favourable results obtained in [23,30] when applying the modified potential fields (PFs) (for distances of several kilometres), if the detection range is limited ($d_{range} = 200$ m [59]) the PFs [23,30] do not provide acceptable results. As a secondary contribution of this work, and based on the concept of generalized potential fields (GPFs) proposed in [45] and studied in [46], we propose to use potential fields that consider the relative direction between the obstacle and the vessel's velocity vector, see Equation (29). For this purpose, the module of repulsive forces ($f_{rep}^{i_L}$) calculated in (26) is replaced by:

$$f_{repG}^{i_L} = f_{rep}^{i_L} |\cos(\phi^{i_L})^{\beta_{gpf}}|, \quad \phi^{i_L} = \begin{cases} \pi/2 & \text{if } |i_L h_{res} - \zeta(k)| > \pi/2 \\ i_L h_{res} - \zeta(k) & \text{otherwise} \end{cases} \quad (29)$$

where β_{gpf} is a positive tuning parameter that decreases the effect of obstacles towards which the USV does not go.

The attraction force in the GPFs method is calculated as in the PFs proposed in [23]. Therefore, any change in the guidance direction of one method with respect to another depends only on the repulsive

forces. A comparative between the repulsive forces generated when applying the PFs proposed in [23,30] and those obtained when applying the GPFs proposed in this paper is shown in Figure 9. As can be seen in the bottom left graph, in the case of GPFs, only the obstacles towards which the USV is moving produce a repulsive force (blue circles in Figure 9). While, in the case of PFs, all obstacles under the range of influence (p_o) produce repulsive forces (brown circles in Figure 9). As a consequence, at the simulation instant shown, the resultant force obtained when applying GPFs ($F_{t_{GPF}}$) guides the USV towards the goal, while if the resultant force obtained by the PFs ($F_{t_{PF}}$) were applied, this would drive the USV away from the obstacle zone (although these do not represent a real danger) and, consequently, away from the goal (see bottom right graph). In the simulation shown, the GPFs method is the one that guides the USV, which has been manually adjusted in an iterative procedure.

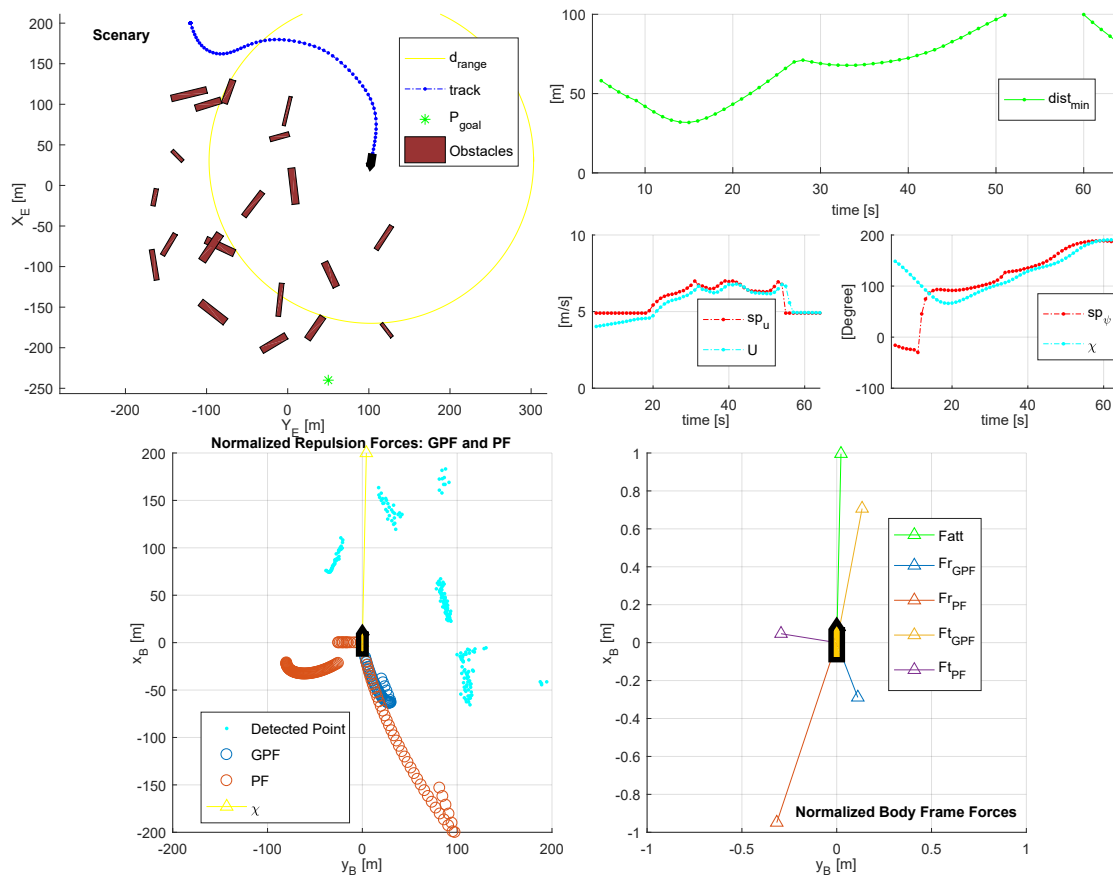


Figure 9. Implementation of the potential fields, as obstacle avoidance algorithms, adapted to the simulation environment for USVs proposed in this work.

3.3. VFH+ Method

The VFH+ method is proposed in [67] as an improved version of the VFH (Vector Field Histogram) [68]. Later, it has been used as an obstacle avoidance system for an USV in [10]. In this work, the block of this algorithm available in Simulink is used [69]. As model of the environment, the VFH+ method uses The Primary Polar Histogram (H_p), [67]. To generate it, the version of the method available in [69] is fully compatible with the model of the LIDAR sensor proposed in this work. As inputs, this block uses the distance measurements (14) and their angular positions (30). Furthermore, as in the other SOA methods, and in order to consider the effect of environmental disturbances [5,52], in this work the angular positions of the measured distances are referenced to the course angle $\chi(k)$.

$$\theta_m^L = \left(\theta_m^0 \quad \dots \quad \theta_m^{i_L} \quad \dots \quad \theta_m^{n_L-1} \right)^T, \quad \theta_m^{i_L} = i_L h_{res} - \text{atan2}(v(k), u(k)) \quad (30)$$

In the same way that the LROABRA method increases the angular occupation of obstacles with the parameter σ , and in contrast to the potential fields which do not consider the size of the USV, the VFH+ method approximates the shape of the vehicle to a circle of radius r_{width} . Thus, the angular occupation of each detected point is extended according to $\gamma_r^{i_L}$, see Equation (31). In this way, each bar of \mathbf{H}^P is defined according to the Equation (32).

$$\gamma_r^L = \left(\gamma_r^0 \quad \dots \quad \gamma_r^{i_L} \quad \dots \quad \gamma_r^{n_L-1} \right)^T, \quad \gamma_r^{i_L} = \begin{cases} \text{asin} \left(r_{width} / d_m^{i_L} \right) & \text{if } f_{Inter}^{i_L} = 1 \\ 0 & \text{otherwise} \end{cases} \quad (31)$$

$$\mathbf{H}^P = \left(h_P^0 \quad \dots \quad h_P^{i_L} \quad \dots \quad h_P^{n_L-1} \right)^T, \quad h_P^{i_L} = \sum_{i_L=1}^{n_O} f \left(d_m^{i_L} \right) b^{i_L} \quad (32)$$

$$b^{i_L} = \begin{cases} 1 & \text{if } i_L h_{res} \in [\theta_m^{i_L} - \gamma_r^{i_L}, \theta_m^{i_L} + \gamma_r^{i_L}] \\ 0 & \text{otherwise} \end{cases}$$

where $f \left(d_m^{i_L} \right)$ defines the occupation density as a function of distance, see [67,69].

Obtained \mathbf{H}^P , this is converted into a Binary Polar Histogram (\mathbf{H}^B) that classifies the directions between occupied or safe. To ensure that the state of the directions does not change several times, which generates an oscillatory behavior that can drive the system to dangerous situations [67], the hysteresis thresholds τ_{high} and τ_{low} are applied:

$$\mathbf{H}^B = \left(h_B^0 \quad \dots \quad h_B^{i_L} \quad \dots \quad h_B^{n_L-1} \right)^T, \quad h_B^{i_L} = \begin{cases} 1 & \text{if } h_P^{i_L}(k) > \tau_{high} \\ 0 & \text{if } h_P^{i_L}(k) < \tau_{low} \\ h_B^{i_L}(k-1) & \text{otherwise} \end{cases} \quad (33)$$

In addition, in order to take into account the dynamic and kinematic of the vehicle, the VFH+ method simplifies its trajectory to circular arcs of radius r_{ship} , [67]. These circular trajectories limit the safe directions, applying a mask to the \mathbf{H}^B and generating The Masked Polar Histogram:

$$\mathbf{H}^M = \left(h_M^0 \quad \dots \quad h_M^{i_L} \quad \dots \quad h_M^{n_L-1} \right)^T, \quad h_M^{i_L} = \begin{cases} 0 & \text{if } h_B^{i_L}(k) = 0 \text{ and } \theta_m^{i_L} \in \{[\phi_r, \chi], [\chi, \phi_l]\} \\ 1 & \text{otherwise} \end{cases} \quad (34)$$

where ϕ_l and ϕ_r are the safe angles to port and starboard, respectively. The definition of these angles is recorded in [67].

Defined \mathbf{H}^M , the course setpoint (sp_ψ) is chosen between the safe or collision-free directions obtained ($h_M^{i_L} = 0$). For this purpose, the following cost function is minimized:

$$g(\theta_m^{i_L}) = \mu_1 \Delta \left(\theta_m^{i_L}, \theta_{goal} \right) + \mu_2 \Delta \left(\theta_m^{i_L}, \chi(k) \right) + \mu_3 \Delta \left(\theta_m^{i_L}, sp_\psi(k-1) \right) \quad (35)$$

where the parameters μ_1 , μ_2 and μ_3 are weighing factors that take into account: alignment with the goal direction, cost associated with the change of direction and alignment with the previous course setpoint (memory effect), respectively. The operator $\delta(c1, c2)$ calculates the angular distance between the directions $c1$ and $c2$.

Finally, the speed setpoint sp_u is calculated according to the Equation (36), [68]. Figure 10 shows one execution of the VFH+ method implemented in this work.

$$sp_u(k) = u_{goal} \left(1 - \frac{|r(k)|}{r_{max}} \right) \quad (36)$$

where r_{max} is a tuning parameter which represents the maximum angular rate of the USV.

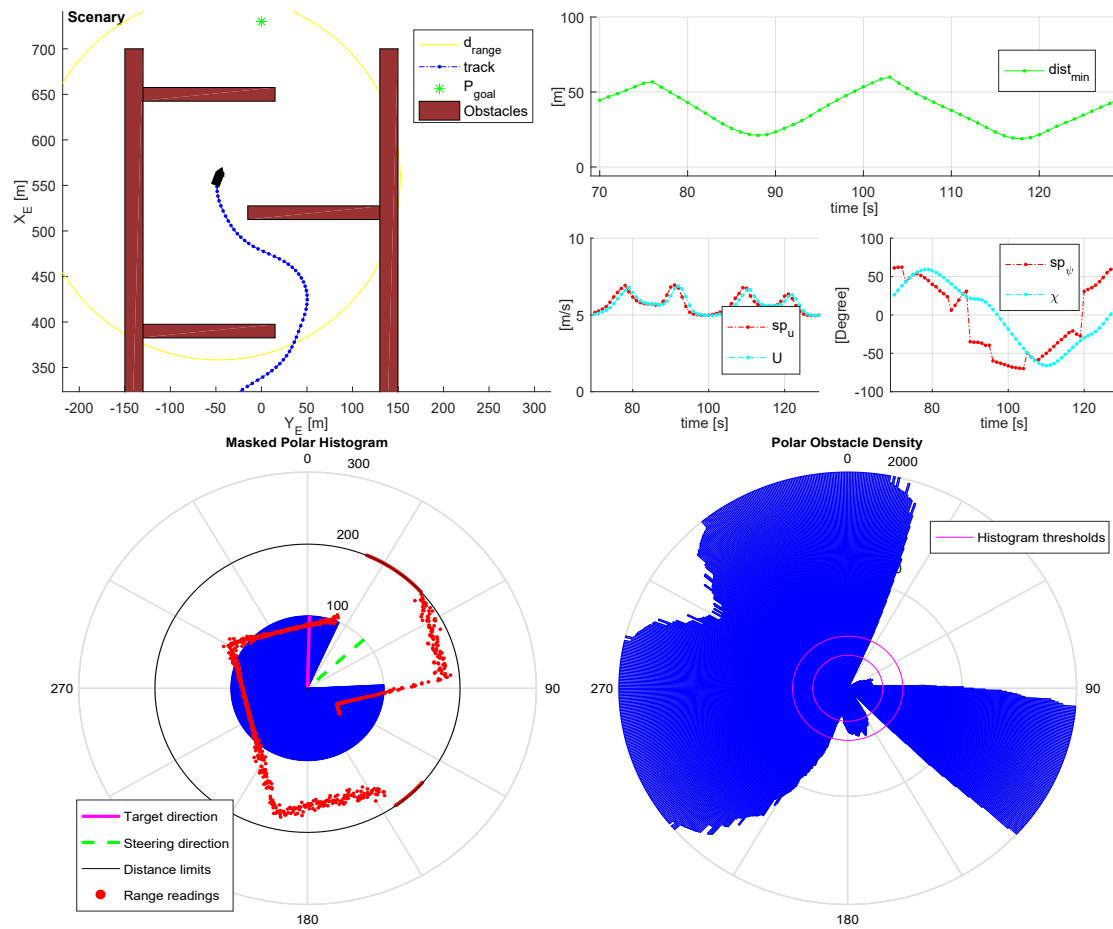


Figure 10. Adaptation of the VFH+ algorithm available in [69] to the simulation environment for USVs proposed in this paper.

4. Simple Autotuning Approach for Obstacle Avoidance Methods

Currently there are a great variety of obstacle avoidance methods for USVs [8,10,12–15,21–31], as well as a wide range of surface marine vehicles [4,5,8,10,12,21,22,26,28,29,47–49,52] in which these can be applied. For this reason, and as main novelty contribution of this work, an AutoTuning Environment for Static Obstacle Avoidance (ATESOA) methods applied to USVs is proposed. This environment is flexible to any type of marine surface vehicle (that is equipped with course and speed controllers) and can be applied to different SOA methods. It should be noted that, starting from the limitations of each avoidance method, its performance to avoid obstacles and to reach the goal are determined by the tuning of its parameters (Θ^{tuning}). In addition, this tuning must be carried out on a specific obstacle scenario. However, in the problems of obstacle avoidance there can be an infinite number of geometric combinations that define the scenario. Therefore, even if the optimum value of Θ^{tuning} is obtained for a defined function, this parameter vector would only be optimal for that particular scenario. For this reason, the study carried out in this work does not pretend to find the optimal parameter vector for each avoidance method applied to a specific vessel. Instead, its purpose is to facilitate the pretuning process necessary to implement SOA methods in USVs, as well as to assist in their choice for a particular vessel, through the use of the new ATESOA proposed in this paper.

In this work, a Genetic Algorithm is used as optimization technique to perform autotuning. This election is due to its established use in vehicle autotuning [37,39,42,44] and to the fact that the methods of obstacle avoidance [8,10,12–15,21–31] usually include decision factors based on thresholds (that convert them into non-smooth functions, complicating the application of conventional least-squares optimization methods [43,53]). Thus, each tuning parameter of an avoidance method is considered as a

gene and, consequently, its tuning parameter vector Θ^{tuning} represents an individual of the population. In addition, to quantify the performance achieved by an avoidance method, with a particular tuning vector Θ^{tuning} , the following indicators are proposed:

$$\begin{aligned}\Delta sp_{\psi}(k) &= \frac{|sp_{\psi}(k) - sp_{\psi}(k-1)|}{\pi} + \Delta sp_{\psi}(k-1) \\ \Delta sp_u(k) &= \frac{|sp_u(k) - sp_u(k-1)|}{u_{max}} + \Delta sp_u(k-1) \\ dist_D(k) &= |(x_E(k) - x_E(k-1)), (y_E(k) - y_E(k-1))| + dist_D(k-1) \\ time(k) &= kT_m^A \\ dist_G(k) &= |(x_E(k) - x_{goal}), (y_E(k) - y_{goal})|\end{aligned}\quad (37)$$

where Δsp_{ψ} and Δsp_u are used to quantify the control effort solicited by the avoidance system, $dist_D$ is the distance traveled by the USV, $time$ is the travel time and $dist_G$ is the distance up to P_{goal} .

Once the indicators (37) have been defined, the fitness function that quantifies the performance of an avoidance method for each individual Θ^{tuning} is defined according to:

$$J = \begin{cases} J_s = \frac{time(k)}{time^{lim}} + \frac{dist_D(k)}{dist_D^{lim}} + \frac{1}{2} \left(\frac{\Delta sp_{\psi}(k)}{\Delta sp_{\psi}^{lim}} + \frac{\Delta sp_u(k)}{\Delta sp_u^{lim}} \right) & \text{if } f_{stop} = 1 \\ J_f = 3 + \frac{dist_G(k)}{dist_G(0)} & \text{otherwise} \end{cases}\quad (38)$$

where $time^{lim}$, $dist_D^{lim}$, Δsp_{ψ}^{lim} and Δsp_u^{lim} set the maximum allowed values for the indicators (37) in a particular scenario and $dist_G(0)$ represents the initial distance between the USV and P_{goal} . On the other hand, f_{stop} defines two situations: the USV has reached its target or not,

$$f_{stop} = \begin{cases} 1 & \text{if } dist_G(k) \leq d_{wp} \\ 0 & \text{if } (n(k) = n^{lim} \vee \min(D_m^L) \leq d_{collision}) \end{cases}, \quad n \in \{time, dist_D, \Delta sp_{\psi}, \Delta sp_u\} \quad (39)$$

where d_{wp} and $d_{collision}$ represent the distance thresholds, which define whether the USV has reached P_{goal} or has collided, respectively.

In this way, if the parameter vector Θ^{tuning} drives the USV to the goal, within the established limits (39), the index J_s is used to minimize: time, distance and control effort in the travel, see Equation (38). On the other hand, if the safety distance ($d_{collision}$) or any of the established limits is exceeded, the numerical simulation is stopped and the index used is J_f . The value of J_f will always be higher than that of J_s and, furthermore, it depends on the final distance between the USV and P_{goal} . This last facilitates that, if in the initial populations a parameter vector that successfully guides the USV to P_{goal} is not obtained, the GA prioritizes those vectors Θ^{tuning} that drive the USV closest to the goal. Once the fitness function (38) has been described, the tuning parameter vectors of the four SOA methods studied in this work are defined in the Equation (40).

$$\begin{aligned}\Theta_{LROABRA}^{tuning} &= \left(\dot{u}_{max} \quad \dot{r}_{max} \quad \epsilon \quad D_{insert} \quad (d_{near} - D_{insert}) \quad \sigma \quad \theta_{gate} \quad r_{max} \quad \theta_v \right)^T \\ \Theta_{PF}^{tuning} &= \left(m_{att} \quad \alpha_{att} \quad \eta_{rep} \quad n_{rep} \quad p_0 \quad \tau_{FR} \right)^T, \quad \Theta_{GPF}^{tuning} = \left(\Theta_{PF}^{tuning^T} \quad \beta_{GPF} \right)^T \\ \Theta_{VFH+}^{tuning} &= \left(\tau_{low} \quad (\tau_{high} - \tau_{low}) \quad r_{ship} \quad \mu_1 \quad \mu_2 \quad \mu_3 \quad r_{max} \right)^T\end{aligned}\quad (40)$$

For the implementation of the AG, the Toolbox available in Matlab [70] has been used. In order to take into account the physical or adjustment limits established for the vectors Θ^{tuning} , these have been limited considering the guidelines exposed in [23,29,30,66–69], see Equation (41). Consequently, the option *mutationadaptfeasible* has been established as a mutation function [70] and the initial population

has been limited to x^{low} and x^{high} , see Equation (41). On the other hand, in order to limit the simulation times, we have established: a population of 200 individuals, a maximum of 20 generations, and the parameter *MaxStallGenerations* as 25% of the maximum generations. Finally, with the purpose of avoiding early convergence [37,70], elite individuals have been limited to 1.5% of the population and the crossover factor has been decreased (thus increasing the mutation factor) to 0.7. The other parameters are the ones established by default in [70].

$$\begin{aligned} \Theta_{PF}^{low} &= (0.1 \quad 50 \quad 8000 \quad 0.1 \quad 50 \quad 1)^T, \quad \Theta_{PF}^{high} = (5 \quad 1000 \quad 2.0 \times 10^5 \quad 5 \quad 200 \quad 6)^T \\ \Theta_{LROABRA}^{low} &= (0.2 \quad 0.3 \quad 0.5 \quad 60 \quad 0 \quad \pi/18 \quad \pi/4 \quad 0.1 \quad \pi/36)^T, \quad \Theta_{GPF}^{low} = (\Theta_{PF}^{lowT} \quad 2)^T \\ \Theta_{LROABRA}^{up} &= (2 \quad 2.5 \quad 0.9 \quad 100 \quad 100 \quad \pi/6 \quad 3\pi/4 \quad 1 \quad \pi/6)^T, \quad \Theta_{GPF}^{high} = (\Theta_{PF}^{highT} \quad 15)^T \\ \Theta_{VFH+}^{low} &= (50 \quad 10 \quad 20 \quad 10 \quad 0 \quad 0 \quad 0.1)^T, \quad \Theta_{VFH+}^{up} = (500 \quad 1000 \quad 100 \quad 20 \quad 10 \quad 10 \quad 0.5)^T \end{aligned} \quad (41)$$

where the respective magnitudes are expressed in the international system of units.

5. Results

This section shows the results obtained by the new ATESOA applied to the USV model proposed in this work. For numerical simulations, Runge-Kutta numerical integration method of fourth order is used with an integration step of 0.01 s and the SOA methods have a sampling period of $T_m^a = 1$ s. Specifically, four SOA methods (LROABRA, PF, GPF and VFH+) have been automatically adjusted for the USV formed by the vessel Model (1), which is governed by the Controllers (7). In addition, in order to expose the algorithms to a higher level of uncertainty in the environment, these have been adapted to the measurements provided by the LIDAR sensor Model (10) proposed in this work, see Section 3. On the other hand, the scenarios 1, 2 and 3, which are defined in Section 2.4, have been used to carry out the autotuning of the SOA methods. As a result, the parameter vectors of each obstacle avoidance method (automatically adjusted for a scenario following the procedure given in Section 4) are shown in the Table 4. Where, for example, PF^{S1} represents the PF method autotuning for the first scenario.

In order to validate the proposed autotuning environment and to compare the performance of each obstacle avoidance method studied in this work, Figure 11 and Table 5 are presented. On the one hand, Figure 11 shows the different trajectories described by the USV, on the three scenarios of autotuning, when each avoidance method is applied. In addition, Table 5 presents the Indicators (37) that define the performance achieved by each avoidance method when guiding the USV. As can be seen, with the exception of the potential fields applied to scenario 2, the new ATESOA achieves parameter vectors that guide the USV safely to P_{goal} . A situation like scenario 2 is not evaluated by the authors in [23,30]. Since, in the most similar scenario, the authors in [23] guide the model of the Mariner class vessel (171.8-meter vessel) proposed in [52] by a scenario where the space between obstacles is about 10 km, compared with the Model (1) of 9.2 m length and the 50 m of distance between obstacles defined in this paper. Furthermore, in [23], the goal point is contained between the obstacles, which does not cause the F_{att} force to drive the USV into these obstacles (situation posed in Scenario 2). Additionally, with manual tuning, the PFs and GPFs methods have also failed in this scenario. This situation corresponds to the problem described in [19,71], in which the restriction of potential fields when passing through small passages is exposed. To solve it, it would be necessary to apply more advanced potential fields as the one proposed in [71]. On the other hand, the contribution made in this work through the application of the concept of generalized potential fields [45,46] to the improved PFs for USV proposed in [23,30], notably improves the guidance of the USV in the studied scenarios. This can be seen in scenarios 1 and 3, where travel times have been reduced to 54.13% and 58.85% and distances driven to 51.33% and 59.31%, respectively. Nevertheless, the performance obtained, for the measurement range $d_{range} = 200$ m and the USV Model (1, 7), by the methods based on potential fields is significantly lower than that obtained by the LROABRA and VFH+ methods, see Table 5. With

respect to these, both present very similar indicators in the three scenarios. The performance obtained by the VFH+ is better in scenarios 1 and 2, while in scenario 3 the LROABRA method is superior.

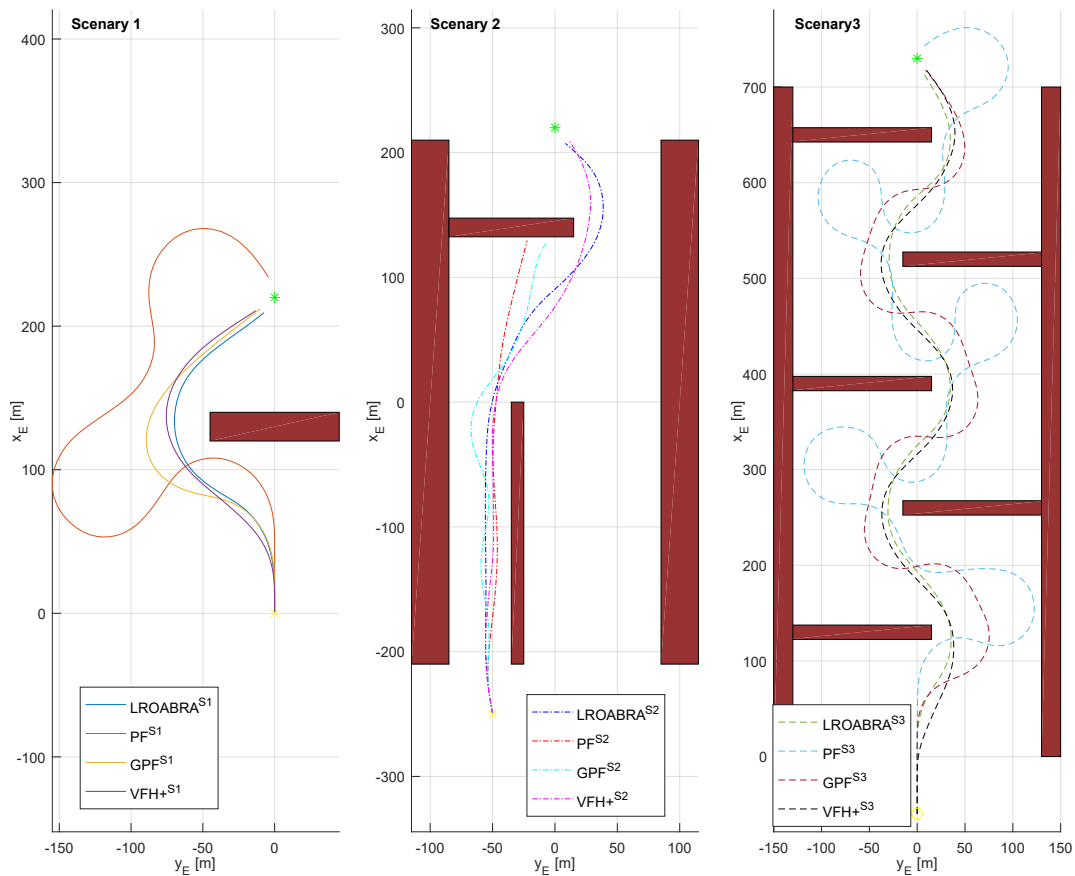


Figure 11. Trajectories of the USV ((1), (7)) by applying each autotuning obstacle avoidance method (LROABRA, PF, GPF and VFH+) for scenarios 1, 2 and 3 with the ATESOA proposed in this work.

On the other hand, in the problems of obstacle avoidance there are infinite geometric combinations that define the possible scenarios that the autonomous vehicle may encounter. Therefore, once the tuning of an avoidance method for a specific scenario has been made, its robustness for other scenarios with different obstacle distributions must be evaluated. In this aspect, a robust behaviour in the guidance of the USV, which is realized by an avoidance method, that keeps its performance (based on the Indicators (37) or similar) is not the objective. Instead, the capability of the avoidance method to safely guide the USV to P_{goal} in significantly different scenarios than those used for its tuning is analyzed. In this context, each avoidance method has been evaluated, with the parameter vectors collected in Table 4, in scenarios with different obstacle distributions than those used for their autotuning. As a result, Figure 12 shows the paths described by the USV when applying each avoidance method over the scenarios defined in this work for which it has not been adjusted. In addition, as support to the previous figure, Table 6 shows the Indicators (37) achieved by each avoidance method in the guidance of the USV. In this robustness study, the indicator that presents the greatest interest is f_{stop} , which defines success or failure in the vehicle's guidance. As can be seen, in the scenarios defined in this work, the PFs and GPFs methods do not present good results in the guidance of the USV over scenarios for which these have not been adjusted. Since only a favourable result is obtained by the GPFs method adjusted in Scenario 3 when it guides the USV over Scenario 1 (GPF^{S3}-S1). In the other scenarios, the methods based on potential fields lead the USV to a collision or generate trajectories that are proper of a limit cycle (PF^{S1} in Scenarios 3 and 5). On the other hand, the LROABRA method successfully solves the majority of scenarios for which it has not been adjusted. With two exceptions:

Scenario 2 with the autotuning performed for scenario 1 and scenario 4 with any parameter vector. The first case is due to the value $d_{near} = 141.72$ m obtained in the autotuning, which constrains the passage between the two parallel obstacles. As a consequence, the avoidance method, without taking into account the USV's dynamics, requires that the vessel turn completely to starboard, which causes a collision. In the second case, the numerical simulations carried out show that, in scenario 4, the collisions occur because the LROABRA method makes multiple switches in the guidance waypoint P_{insert} . These commutations, as established in [67], cause undecided behaviour that leads the USV to a collision. To correct this, in a similar way to the VFH+ method, a possible solution would consist in adding to the heuristics (19) a weighting factor that takes into account the alignment of the candidate directions V_{Head} with the previous course setpoint $sp_{\psi}(k-1)$.

Table 4. Parameter vectors obtained by ATESOA in the scenarios 1, 2 and 3.

Method ^{Scenario}	Parameter Vectors
LROABRA ^{S1}	$\Theta^{tuning} = (1.83 \quad 1.87 \quad 0.68 \quad 62.28 \quad 79.44 \quad 0.19 \quad 0.86 \quad 0.77 \quad 0.51)^T$
LROABRA ^{S2}	$\Theta^{tuning} = (1.43 \quad 1.84 \quad 0.63 \quad 67.13 \quad 41.91 \quad 0.19 \quad 1.17 \quad 0.66 \quad 0.11)^T$
LROABRA ^{S3}	$\Theta^{tuning} = (1.07 \quad 1.57 \quad 0.56 \quad 60.98 \quad 60.35 \quad 0.18 \quad 2.18 \quad 0.68 \quad 0.09)^T$
PF ^{S1}	$\Theta^{tuning} = (1.36 \quad 833.16 \quad 1.32 \times 10^5 \quad 1.68 \quad 107.95 \quad 1.54)^T$
PF ^{S2}	$\Theta^{tuning} = (2.06 \quad 550.53 \quad 2.02 \times 10^4 \quad 1.75 \quad 100.59 \quad 1.50)^T$
PF ^{S3}	$\Theta^{tuning} = (2.85 \quad 454.25 \quad 1.03 \times 10^5 \quad 4.20 \quad 172.67 \quad 1.91)^T$
GPF ^{S1}	$\Theta^{tuning} = (2.62 \quad 909.49 \quad 1.89 \times 10^5 \quad 4.48 \quad 161.7 \quad 2.27 \quad 14.49)^T$
GPF ^{S2}	$\Theta^{tuning} = (1.07 \quad 245.22 \quad 1.85 \times 10^5 \quad 0.34 \quad 141.87 \quad 3.28 \quad 10.64)^T$
GPF ^{S3}	$\Theta^{tuning} = (0.75 \quad 447.15 \quad 1.55 \times 10^5 \quad 0.50 \quad 120.94 \quad 1.06 \quad 13.75)^T$
VFH+ ^{S1}	$\Theta^{tuning} = (93.71 \quad 714.1 \quad 50.94 \quad 15.81 \quad 7.58 \quad 0.30 \quad 0.45)^T$
VFH+ ^{S2}	$\Theta^{tuning} = (305.97 \quad 176.25 \quad 72.96 \quad 13.01 \quad 2.12 \quad 9.71 \quad 0.43)^T$
VFH+ ^{S3}	$\Theta^{tuning} = (407.43 \quad 40.24 \quad 70.08 \quad 19.84 \quad 3.13 \quad 3.18 \quad 0.47)^T$

Table 5. Indicators (37) achieved by each obstacle avoidance method in the scenarios 1, 2 and 3.

Method-Scenario/Indicators	f_{stop}	time [s]	dist _D [m]	Δsp_{ψ} [rad]	Δsp_u [m/s]	dist _G [m]
LROABRA ^{S1} - S1	✓	50	263.64	248.02	1.08	—
PF ^{S1} - S1	✓	109	573.20	446.68	2.33	—
GPF ^{S1} - S1	✓	59	294.23	427.20	1.11	—
VFH+ ^{S1} - S1	✓	47	265.50	162.14	1.29	—
LROABRA ^{S2} - S2	✓	82	751.45	154.19	1.74	—
PF ^{S2} - S2	✗	—	—	—	—	93.19
GPF ^{S2} - S2	✗	—	—	—	—	91.62
VFH+ ^{S2} - S2	✓	78	739.36	131.86	1.23	—
LROABRA ^{S3} - S3	✓	145	929.25	340.32	2.85	—
PF ^{S3} - S3	✓	384	1993.10	1785.40	6.92	—
GPF ^{S3} - S3	✓	226	1182.10	1837.20	1.05	—
VFH+ ^{S3} - S3	✓	148	949.69	428.77	2.28	—

With respect to the VFH+ method, it successfully resolves all of the scenarios defined in this paper for which it has not been adjusted. Moreover, it presents a robust behaviour to the variations in its

parameter vector, since the Indicators (37) obtained in the same scenario for different values of Θ^{tuning} are very close, see Table 6. Therefore, for the defined scenarios, of the four methods of static obstacle avoidance applied to USVs that have been adapted and implemented in this work, the VFH+ method achieves the best results in terms of performance (see Table 5) and robustness (see Figure 12).

Table 6. Indicators (37) achieved in the guidance of the USV by the SOA methods (LROABRA, PF, GPF and VFH+) in scenarios different to those used for their autotuning.

Method-Scenario/Indicators	f_{stop}	time [s]	$dist_D$ [m]	Δsp_ψ [rad]	Δsp_u [m/s]	$dist_G$ [m]
LROABRA ^{S2} - S1	✓	71	350.42	573.68	1.67	—
LROABRA ^{S3} - S1	✓	69	346.06	401.13	1.54	—
LROABRA ^{S1} - S2	✗	—	—	—	—	408.43
LROABRA ^{S3} - S2	✓	82	752.03	133.03	1.65	—
LROABRA ^{S1} - S3	✓	167	951.04	509.81	3.83	—
LROABRA ^{S2} - S3	✓	165	1002.80	638.77	3.83	—
LROABRA ^{S1} - S4	✗	—	—	—	—	362.89
LROABRA ^{S2} - S4	✗	—	—	—	—	357.68
LROABRA ^{S3} - S4	✗	—	—	—	—	160.49
LROABRA ^{S1} - S5	✓	109	780.98	344.93	1.60	—
LROABRA ^{S2} - S5	✓	89	767.74	236.10	2.05	—
LROABRA ^{S3} - S5	✓	88	758.31	232.56	2.01	—
PF ^{S3} - S1	✗	—	—	—	—	103.32
PF ^{S1} - S2	✗	—	—	—	—	419.99
PF ^{S3} - S2	✗	—	—	—	—	406.31
PF ^{S1} - S3	✗	—	—	—	—	646.32
PF ^{S1} - S4	✗	—	—	—	—	422.80
PF ^{S3} - S4	✗	—	—	—	—	418.08
PF ^{S1} - S5	✗	—	—	—	—	303.24
PF ^{S3} - S5	✗	—	—	—	—	139.99
GPF ^{S3} - S1	✓	64	324.85	356.46	1.14	—
GPF ^{S1} - S2	✗	—	—	—	—	323.60
GPF ^{S3} - S2	✗	—	—	—	—	360.60
GPF ^{S1} - S3	✗	—	—	—	—	222.95
GPF ^{S1} - S4	✗	—	—	—	—	226.71
GPF ^{S3} - S4	✗	—	—	—	—	228.07
GPF ^{S1} - S5	✗	—	—	—	—	181.70
GPF ^{S3} - S5	✗	—	—	—	—	177.98
VFH+ ^{S2} - S1	✓	48	270.62	164.54	1.33	—
VFH+ ^{S3} - S1	✓	47	267.96	220.53	1.27	—
VFH+ ^{S1} - S2	✓	80	748.90	130.65	1.39	—
VFH+ ^{S3} - S2	✓	78	743.54	129.58	1.21	—
VFH+ ^{S1} - S3	✓	177	1081.70	644.07	2.96	—
VFH+ ^{S2} - S3	✓	166	1018.60	600.59	3.39	—
VFH+ ^{S1} - S4	✓	78	758.63	194.63	1.44	—
VFH+ ^{S2} - S4	✓	76	754.82	157.85	1.30	—
VFH+ ^{S3} - S4	✓	75	750.52	151.35	1.23	—
VFH+ ^{S1} - S5	✓	93	791.22	400.88	1.96	—
VFH+ ^{S2} - S5	✓	92	790.23	275.57	1.83	—
VFH+ ^{S3} - S5	✓	92	793.59	287.66	1.84	—

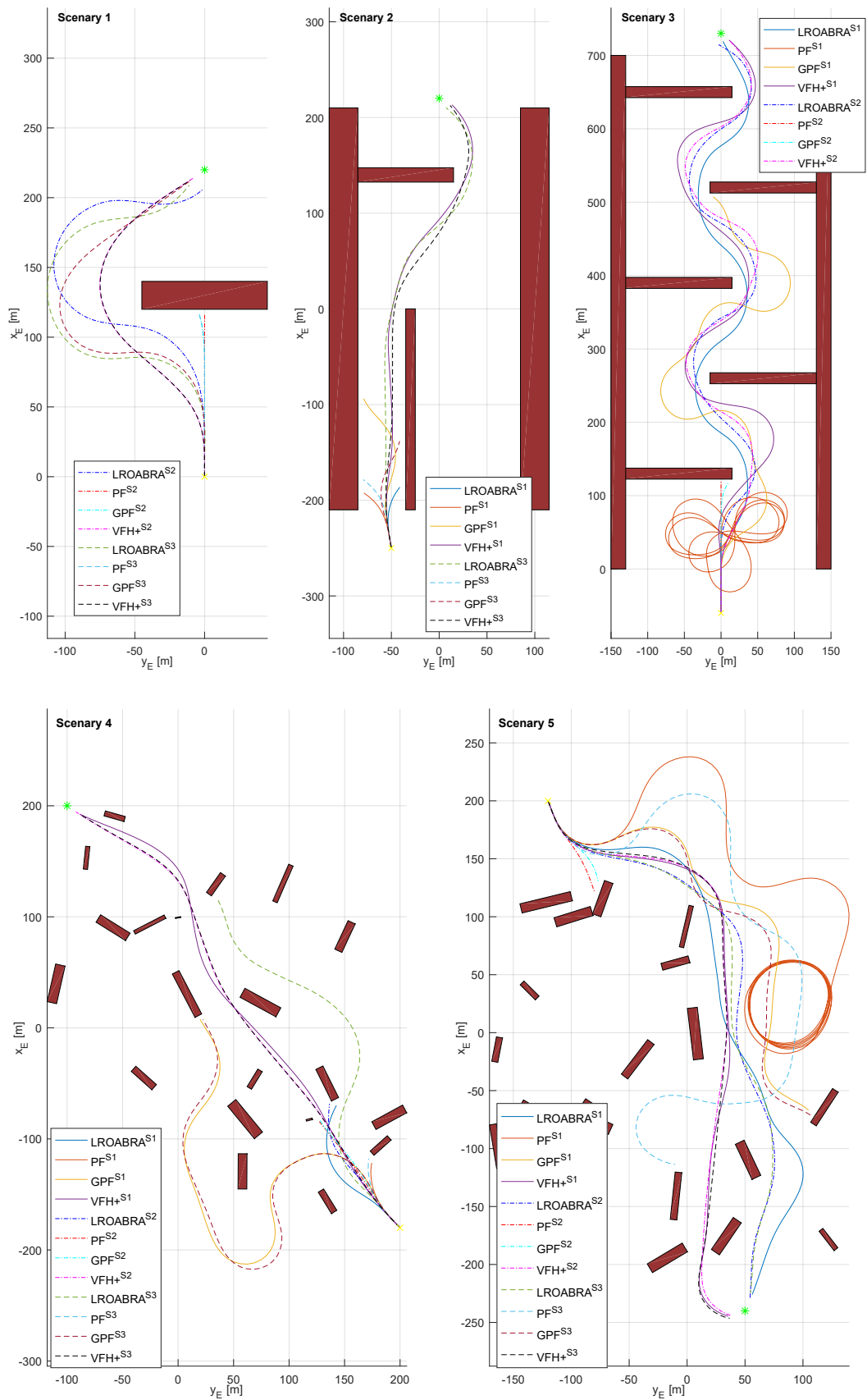


Figure 12. Trajectories of the USV ((1), (7)) when applying the LROABRA, PF, GPF and VFH+ methods in different scenarios than those used for its autotuning.

As shown in the results obtained in Figure 12 and in Table 6, the new ATESOA proposed in this work is flexible, such that it can be applied to different SOA methods used in USVs. Where, the robustness of the autotuning obtained depends on the scenario used, but mainly of the features specific of each obstacle avoidance method. Since, as shown below, the new ATESOA achieves a successful adjustment of the SOA methods that did not overcome the obstacle scenarios 4 and 5. In particular, the autotuning of the LROABRA method for scenario 4 and of the methods based on potentials fields for scenarios 4 and 5 is carried out. The parameter vectors obtained are collected in Table 7. As a result, Figure 13 shows the trajectories described by the USV when applying the LROABRA, PFs and GPFs methods autotuning in the new ATESOA. In addition, Table 8 presents the Indicators (37) obtained by each method in each scenario. As can be seen, all methods successfully solve the two scenarios. In scenario 4, the LROABRA method again improves performance with respect to potential fields. Regarding the potential fields [23,30], of the methods studied in this work, the results obtained indicate that these would be the worst option as an avoidance system for USVs in situations where the measurement range of the sensors is limited ($d_{range} \leq 200$ m). Although in both scenarios, the GPFs method proposed in this work again produces a less oscillatory trajectory in the guidance than the PFs proposed in [23,30], which means shorter travel distances.

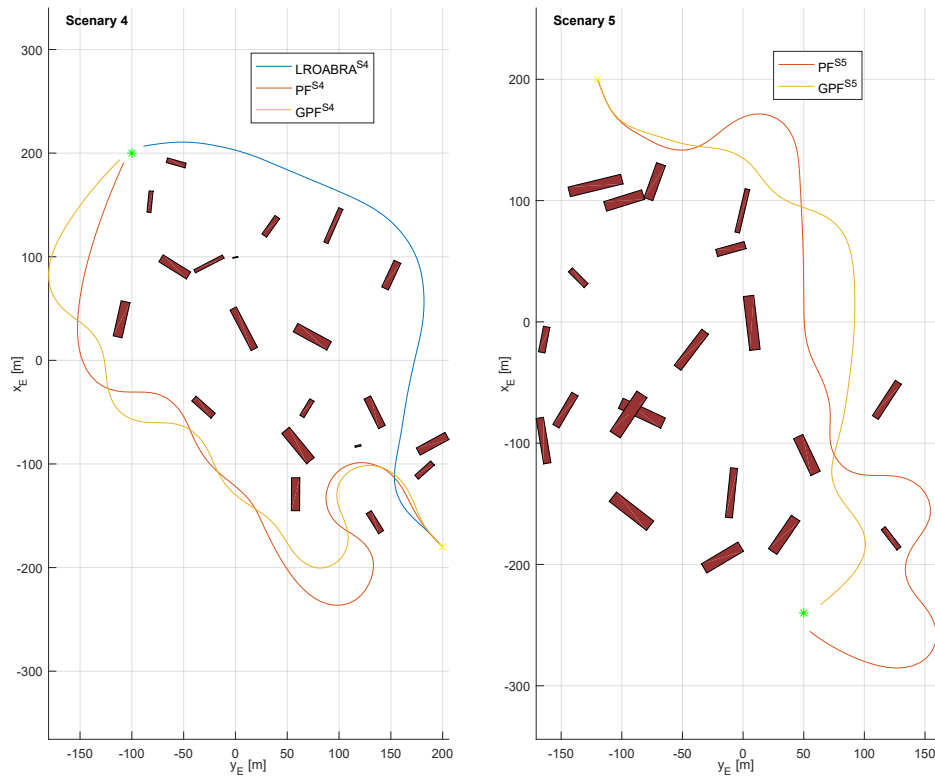


Figure 13. Trajectories of the USV ((1), (7)) by applying the methods LROABRA, PFs and GPFs autotuning.

Table 7. Parameter vectors obtained for the LROABRA, PF and GPF methods, scenarios 4 and 5.

Method ^{Scenario}	Parameter Vectors
LROABRA ^{S4}	$\Theta^{tuning} = (1.92 \quad 0.83 \quad 0.56 \quad 74.94 \quad 74.91 \quad 0.21 \quad 1.41 \quad 0.85 \quad 0.11)^T$
PF ^{S4}	$\Theta^{tuning} = (1.97 \quad 209.81 \quad 1.10 \times 10^5 \quad 3.24 \quad 34.02 \quad 5.62)^T$
PF ^{S5}	$\Theta^{tuning} = (0.99 \quad 525.02 \quad 5.46 \times 10^4 \quad 0.73 \quad 42.79 \quad 4.09)^T$
GPF ^{S4}	$\Theta^{tuning} = (1.96 \quad 867.47 \quad 5.34 \times 10^4 \quad 3.31 \quad 76.05 \quad 3.92 \quad 11.45)^T$
GPF ^{S5}	$\Theta^{tuning} = (0.70 \quad 726.56 \quad 1.70 \times 10^5 \quad 1.73 \quad 29.30 \quad 5.99 \quad 10.66)^T$

Table 8. Indicators (37) achieved by the LROABRA, PF and GPF methods in the scenarios 4 and 5.

Method-Scenario/Indicators	f_{stop}	time [s]	$dist_D$ [m]	Δsp_ψ [rad]	Δsp_u [m/s]	$dist_G$ [m]
LROABRA ^{S4} - S4	✓	97	867.66	290.04	1.12	—
PF ^{S4} - S4	✓	144	1143.20	511.33	4.36	—
GPF ^{S4} - S4	✓	153	1063.40	915.83	2.04	—
PF ^{S5} - S5	✓	137	1047.60	642.90	3.70	—
GPF ^{S5} - S5	✓	95	815.77	443.22	2.67	—

6. Conclusions

Due to the scientific and industrial interest, in the last years many works have been developed in order to provide a higher level of autonomy to the vessels. This has led to a diversity of methods for static obstacle avoidance applied to USVs. In order to design, development and validation of these methods, it is necessary an environment for computer simulation that allows the evaluation of the proposed algorithms. With this aim, the main contribution of this work is the new ATESOA, an autotuning environment for the design parameters associated with each static obstacle avoidance algorithm. This environment is adaptable to diverse SOA methods, which can be applied in different types of USVs. Due to the simplified model of the LIDAR sensor proposed in this paper, the simulation environment is more realistic than those used by most authors for the design of obstacle avoidance methods. With the advantage, over works focused on more sophisticated and complex simulation environments to validate USVs, that this model of the LIDAR sensor facilitates the application of evolutionary algorithms for autotuning (due to its lower computation load). To evaluate the new ATESOA, four SOA methods (LROABRA, PFs, GPFs and VFH+) have been adapted and implemented for using this sensor model. The results show how this new autotuning environment achieves a successful and, depending on the obstacle avoidance method, robust tuning of the parameters of the SOA methods for a range of measurement limited by the LIDAR [59] to 200 m. In addition, the parameters for the mathematical model of a USV of 9.2 m of length, whose modelling is based on previous work in the field of surface marine vehicle control, are proposed and evaluated. This model includes: non-linear modelling of the vessel and actuators dynamics, modelling of the effect of ocean currents and course/velocity controllers. In addition, as method of avoiding reactive obstacles, the performance of potential fields [23,30], applied to this USV model (length of 9.2 m) has been improved, through the adaptation of the generalized potential fields [45,46]. As future work, the new autotuning environment will be adapted for dynamic obstacle avoidance methods.

Author Contributions: R.G. was responsible for studying, adapting and implementing the obstacle avoidance methods, the LIDAR sensor model and the new autotuning environment. In addition, R.G. wrote most of the work. M.J.L. proposed and tested the new mathematical modelling of USV proposed in this work. M.J.L. has also participated in the analysis of proposals, methods and discussion of results. In addition, M.J.L. wrote part of the paper and reviewed in depth all its contents. J.S., due to his experience with the USVs in the Navantia company, reviewed all the work and, more in detail, the vessel's model, the course/speed controllers and the obstacle avoidance methods. Finally, A.C. reviewed the entire article and contributed to its translation. All authors have read and agreed to the published version of the manuscript.

Funding: This research has been carried out within the collaborative framework of industrial doctoral thesis, which is financed by the Universidad de Cádiz and the company Navantia.

Acknowledgments: The development and financing of this work has been possible thanks to the vision and passion of Alberto González Cantos, a senior USV professional at Navantia. In addition, it is also necessary to thank the University of Cadiz for the funds to finance this paper. As well as the research group GAPSIS (Grupo de Automática, Procesamiento de Señales e Ingeniería de Sistemas), whose previous work has allowed the formulation of the framework in which this work has been carried out.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, Z.; Zhang, Y.; Yu, X.; Yuan, C. Unmanned surface vehicles: An overview of developments and challenges. *Annu. Rev. Control* **2016**, *41*, 71–93. [\[CrossRef\]](#)
2. Campbell, S.; Naeem, W.; Irwin, G.W. A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annu. Rev. Control* **2012**, *36*, 267–283. [\[CrossRef\]](#)
3. González, D.; Pérez, J.; Milanés, V.; Nashashibi, F. A Review of Motion Planning Techniques for Automated Vehicles. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 1135–1145.
4. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: Trondheim, Norway, 2011; ISBN 9781119991496.
5. Fossen, T.I. *Marine Control Systems*; John Wiley & Sons: Trondheim, Norway, 2002; ISBN 8292356002.
6. Elfes, A. Using Occupancy Grids for Mobile Robot Perception and Navigation. *Computer* **1989**, *22*, 46–57. [\[CrossRef\]](#)
7. Lavalle, S. *Planning Algorithms*; Cambridge University Press: New York, NY, USA, 2006; ISBN 978-0-521-86205-9.
8. Perera, L.P.; Ferrari, V.; Santos, F.P.; Hinostroza, M.A.; Guedes Soares, C. Experimental Evaluations on Ship Autonomous Navigation and Collision Avoidance by Intelligent Guidance. *IEEE J. Ocean. Eng.* **2015**, *40*, 374–387. [\[CrossRef\]](#)
9. Perera, L.P.; Oliveira, P.; Guedes Soares, C. Maritime Traffic Monitoring Based on Vessel Detection, Tracking, State Estimation, and Trajectory Prediction. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 1188–1200. [\[CrossRef\]](#)
10. Kim, T.; Choi, J.; Lee, Y.; Choi, H.-T. VFH+ based Obstacle Avoidance using Monocular Vision of Unmanned Surface Vehicle. *J. Ocean Eng. Technol.* **2016**, *30*, 426–430. [\[CrossRef\]](#)
11. Minguez, J.; Lamiriaux, F.; Laumond, J.-P. Motion Planning and Obstacle Avoidance. In *Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2008; Volume E, pp. 827–852, ISBN 9783540303015.
12. Kuwata, Y.; Wolf, M.T.; Zarzhitsky, D.; Huntsberger, T.L. Safe maritime autonomous navigation with COLREGS, using velocity obstacles. *IEEE J. Ocean. Eng.* **2014**, *39*, 110–119. [\[CrossRef\]](#)
13. Zhao, Y.; Li, W.; Shi, P. A real-time collision avoidance learning system for Unmanned Surface Vessels. *Neurocomputing* **2016**, *182*, 255–266. [\[CrossRef\]](#)
14. Perera, L.P.; Carvalho, J.P.; Guedes Soares, C. Intelligent ocean navigation and fuzzy-Bayesian decision/action formulation. *IEEE J. Ocean. Eng.* **2012**, *37*, 204–219. [\[CrossRef\]](#)
15. Johansen, T.A.; Perez, T.; Cristofaro, A. Ship collision avoidance and COLREGS compliance using simulation-based control behavior selection with predictive hazard assessment. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 3407–3422. [\[CrossRef\]](#)
16. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D. Path Planning of an Autonomous Surface Vehicle based on Artificial Potential Fields in a Real Time Marine Environment. In Proceedings of the 16 International Conference on Computer and IT Applications in the Maritime Industries, Cardiff, Gales, UK, 15–17 May 2017; pp. 48–54.
17. Bibuli, M.; Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D.; Khan, A. A Two Layered Optimal Approach towards Cooperative Motion Planning of Unmanned Surface Vehicles in a Constrained Maritime Environment. *IFAC-PapersOnLine* **2018**, *29*, 378–383. [\[CrossRef\]](#)
18. Gonzalez Cantos, A. *Aplicación de la Teoría Cualitativa de Sistemas Dinámicos para el Guiado Autónomo de Vehículos*; Universidad de Cádiz: Cádiz, Spain, 2017.
19. Singh, Y.; Sharma, S.; Sutton, R.; Hatton, D. Towards use of Dijkstra Algorithm for Optimal Navigation of an Unmanned Surface Vehicle in a Real-Time Marine Environment with results from Artificial Potential Field. *TransNav Int. J. Mar. Navig. Saf. Sea Transp.* **2018**, *12*, 125–131. [\[CrossRef\]](#)
20. da Silva Junior, A.G.; dos Santos, D.H.; de Negreiros, A.P.F.; de Souza Silva Silva, J.M.V.B.; Gonçalves, L.M.G. High-Level Path Planning for an Autonomous. *Sensors* **2020**, *20*, 1550. [\[CrossRef\]](#) [\[PubMed\]](#)
21. Agrawal, P.; Dolan, J.M. COLREGS-compliant target following for an Unmanned Surface Vehicle in dynamic environments. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1065–1070.
22. Abdelaal, M.; Fränzle, M.; Hahn, A. Nonlinear Model Predictive Control for trajectory tracking and collision avoidance of underactuated vessels with disturbances. *Ocean Eng.* **2018**, *160*, 168–180. [\[CrossRef\]](#)

23. Xue, Y.; Lee, B.S.; Han, D. Automatic collision avoidance of ships. *Proc. Inst. Mech. Eng. Part M J. Eng. Marit. Environ.* **2009**, *223*, 33–46. [\[CrossRef\]](#)
24. Zhang, J.; Zhang, D.; Yan, X.; Haugen, S.; Guedes Soares, C. A distributed anti-collision decision support formulation in multi-ship encounter situations under COLREGs. *Ocean Eng.* **2015**, *105*, 336–348. [\[CrossRef\]](#)
25. Naeem, W.; Irwin, G.W.; Yang, A. COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Mechatronics* **2012**, *22*, 669–678. [\[CrossRef\]](#)
26. Mousazadeh, H.; Jafarbiglu, H.; Abdolmaleki, H.; Omrani, E.; Monhaseri, F.; Abdollahzadeh, M.; Mohammadi-Aghdam, A.; Kiapei, A.; Salmani-Zakaria, Y.; Makhsoos, A. Developing a navigation, guidance and obstacle avoidance algorithm for an Unmanned Surface Vehicle (USV) by algorithms fusion. *Ocean Eng.* **2018**, *159*, 56–65. [\[CrossRef\]](#)
27. Wang, X.; Liu, Z.; Cai, Y. The ship maneuverability based collision avoidance dynamic support system in close-quarters situation. *Ocean Eng.* **2017**, *146*, 486–497. [\[CrossRef\]](#)
28. Loe, Ø.A.G. *Collision Avoidance for Unmanned Surface Vehicles*; Norwegian University of Science and Technology: Trondheim, Norway, 2008.
29. Tang, P.; Zhang, R.; Liu, D.; Huang, L.; Liu, G.; Deng, T. Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle. *Ocean Eng.* **2015**, *106*, 128–140. [\[CrossRef\]](#)
30. Xie, S.; Wu, P.; Peng, Y.; Luo, J.; Qu, D.; Li, Q.; Gu, J. The obstacle avoidance planning of USV based on improved artificial potential field. In Proceedings of the 2014 IEEE International Conference on Information and Automation (ICIA), Hailar, China, 28–30 July 2014; pp. 746–751.
31. Tang, P.; Zhang, R.; Liu, D.; Zou, Q.; Shi, C. Research on near-field obstacle avoidance for unmanned surface vehicle based on heading window. In Proceedings of the 2012 24th Chinese Control and Decision Conference, Taiyuan, China, 23–25 May 2012; pp. 1262–1267.
32. Paravisi, M.; Santos, D.H.; Jorge, V.; Heck, G.; Gonçalves, L.M.; Amory, A. Unmanned surface vehicle simulator with realistic environmental disturbances. *Sensors* **2019**, *19*, 1068. [\[CrossRef\]](#) [\[PubMed\]](#)
33. Kermorgant, O. A Dynamic Simulator for Underwater Vehicle-Manipulators. In Proceedings of the International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAP), Bergamo, Italy, 20–23 October 2014; Springer: Heidelberg/Berlin, Germany, 2014; Volume 8810, pp. 25–36.
34. Rohmer, E.; Singh, S.P.N.; Freese, M. V-REP: A versatile and scalable robot simulation framework. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Tokyo, Japan, 3–7 November 2013; pp. 1321–1326.
35. RobotX Simulator. Available online: <https://bitbucket.org/osrf/vrx/src/default/> (accessed on 8 January 2020).
36. Koenig, N.; Howard, A. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 28 September–2 October 2004; Volume 3, pp. 2149–2154.
37. Simon, D. *Evolutionary Optimization Algorithms*; John Wiley & Sons: Hoboken, NJ, USA, 2013; ISBN 978-0-470-93741-9.
38. Xiao, X.; Liu, B.; Warnell, G.; Fink, J.; Stone, P. APPLD: Adaptive Planner Parameter Learning from Demonstration. *arXiv* **2020**, arXiv2004.00116.
39. McGookin, E.W.; Murray-Smith, D.J.; Li, Y.; Fossen, T.I. Ship steering control system optimisation using genetic algorithms. *Control Eng. Pract.* **2000**, *8*, 429–443. [\[CrossRef\]](#)
40. Alme, J.; Breivik, M. Autotuning aspects for dynamic positioning systems. *IFAC Proc. Vol.* **2009**, *42*, 334–339. [\[CrossRef\]](#)
41. Mišković, N.; Vukić, Z.; Barišić, M.; Tovornik, B. Autotuning autopilots for Micro-ROVs. In Proceedings of the 14th Mediterranean Conference on Control and Automation IEEE, Ancona, Italy, 28–30 June 2006; pp. 1–6.
42. Walker, M.; Messom, C.H. A comparison of genetic programming and genetic algorithms for auto-tuning mobile robot motion control. In Proceedings of the 1st IEEE International Workshop on Electronic Design, Test and Applications, Christchurch, New Zealand, 29–31 January 2002; pp. 507–509.
43. Xiao, H.; Wang, S. Auto-tuning PID module of robot motion system. In Proceedings of the 6th Conference on Industrial Electronics and Applications, Beijing, China, 21–23 June 2011; pp. 668–673.
44. Al-Araji, A.S. Development of Kinematic Path-Tracking Controller Design for Real Mobile Robot via Back-Stepping Slice Genetic Robust Algorithm Technique. *Arab. J. Sci. Eng.* **2014**, *39*, 8825–8835. [\[CrossRef\]](#)

45. Krogh, B.H. A Generalized Potential Field Approach to Obstacle Avoidance Control. In *Robotics Research: The Next Five Years and Beyond*; RI/SME: Bethlehem, PA, USA, 1984.
46. Tilove, R.B. Local obstacle avoidance for mobile robots based on the method of artificial potentials. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Cincinnati, OH, USA, 13–18 May 1990; pp. 566–571.
47. Do, K.D.; Jiang, Z.P.; Pan, J. Robust adaptive path following of underactuated ships. *Automatica* **2004**, *40*, 929–944. [\[CrossRef\]](#)
48. Han, J.; Xiong, J.; He, Y.; Gu, F.; Li, D. Nonlinear Modeling for a Water-Jet Propulsion USV: An Experimental Study. *IEEE Trans. Ind. Electron.* **2017**, *64*, 3348–3358. [\[CrossRef\]](#)
49. Skjetne, R.; Smogeli, Ø.N.; Fossen, T.I. A Nonlinear Ship Manoeuvring Model: Identification and adaptive control with experiments for a model ship. *Identif. Control.* **2004**, *25*, 3–27. [\[CrossRef\]](#)
50. Sørensen, A.J. *Marine Control Systems. Propulsion and Motion Control of Ships and Ocean Structures*; Norwegian University of Science and Technology: Trondheim, Norway, 2012.
51. Perez, T.; Blanke, M. Simulation of Ship Motion in Seaway. In *Computer Science*; Technical Report; The University of Newcastle: Callaghan, Australia, 2002; pp. 1–13.
52. Fossen, T.I. *Guidance and Control of Ocean Vehicles*; John Wiley & Sons: Trondheim, Norway, 1994; ISBN 0471941131.
53. Astrom, K.J.; Wittenmark, B. *Adaptive Control*; Addison Whesley: Boston, MA, USA, 1989; ISBN 0201097206.
54. Kallstrom, C.G. *Identification and Adaptive Control Applied to Ship Steering*; Department of Automatic Control, Lund University: Lund, Sweden, 1979.
55. Lewis, E.V. *Principles of Naval Architecture, Vol III—Motions in Waves and Controllability*; SNAME (The Society of Naval Architects and Marine Engineers): Jersey City, NJ, USA, 1989; ISBN 0939773023.
56. López, M. J.; Consegliere, A.; Terrón, J. *Modelado y simulación de sistemas. Aplicación al buque*; Universidad de Cádiz: Cádiz, España, 1997; doi:10.13140/RG.2.1.3400.5843. [\[CrossRef\]](#)
57. American Bureau of Shipping. *Guide for Vessel Maneuverability*; American Bureau of Shipping: Houston, TX, USA, 2006.
58. Rubio, F.R.; López, M.J. *Control Adaptativo y Robusto*, 1st ed.; Universidad de Sevilla: Sevilla, Spain, 1996; ISBN 84-472-0319-0.
59. Velodyne Lidar. *UltraPuck Data Sheet*; Velodyne: Hellyer Ave, SJ, USA, 2019.
60. O'Rourke, J. *Computational Geometry in C*, 2nd ed.; Cambridge University Press: Cambridge, MA, USA, 1997; ISBN 0-521-64976-5.
61. Gobierno de España, Ministerio de Transportes Movilidad y Agenda Urbana. Lineas de Costa. Available online: <http://centrodedescargas.cnig.es/CentroDescargas/catalogo.do?Serie=CAANE> (accessed on 15 April 2020).
62. U. Murat Erdem. Fast Line Segment Intersection. MATLAB Central File Exchange. Retrieved 13 March 2020. Available online: <https://www.mathworks.com/matlabcentral/fileexchange/27205-fast-line-segment-intersection> (accessed on 4 February 2019).
63. Guardeno, R.; López, M.J.; Sánchez, J.; Consegliere, A. LROABRA algorithm for Static Obstacle Avoidance Adapted and Autotuning by ATESOA. Available online: youtube.com/watch?v=ZEDdUCz_3ZI (accessed on 16 April 2020).
64. Fox, D.; Burgard, W.; Thrun, S. The Dynamic Window Approach to Collision Avoidance. *IEEE Robot. Autom. Mag.* **1997**, *4*, 23–33. [\[CrossRef\]](#)
65. Khatib, O. Real-Time Obstacle Avoidance For Manipulators And Mobile Robots. In Proceedings of the IEEE International Conference on Robotics and Automation, St. Louis, MO, USA, 25–28 March 1985; pp. 500–505.
66. Borenstein, J.; Koren, Y. Real-Time Obstacle Avoidance for Fast Mobile Robots. *IEEE Trans. Syst. Man Cybern.* **1989**, *19*, 1179–1187. [\[CrossRef\]](#)
67. Ulrich, I.; Borenstein, J. VFH+: Reliable Obstacle Avoidance for Fast Mobile Robots. In Proceedings of the 1998 IEEE International Conference on Robotics & Automation; Leuven, Belgium, 16–20 May 1998; pp. 1572–1577.
68. Borenstein, J.; Koren, Y. The vector field histogram—Fast obstacle avoidance for mobile robots. *IEEE Trans. Robot. Autom.* **1991**, *7*, 278–288. [\[CrossRef\]](#)
69. MathWorks. Vector Field Histogram. Available online: <https://es.mathworks.com/help/nav/ug/vector-field-histograms.html> (accessed on 23 May 2019).

70. MathWorks. Genetic Algorithm. Available online: <https://es.mathworks.com/help/gads/genetic-algorithm.html> (accessed on 5 July 2019).
71. Tu, K.Y.; Baltes, J. Fuzzy potential energy for a map approach to robot navigation. *Robot. Auton. Syst.* **2006**, *54*, 574–589. [CrossRef]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).